Feasibility Study and Requirements Definition for Cooperative Human-Adaptive Traffic Simulation (CHATS)

Final Report

December 1998

Prepared for
NASA Ames Research Center
Moffett Field, CA 94035-1000
and
NASA Langley Research Center
Hampton, VA 23681
Under Contract Number NAS2-98005
Task Order 1

Prepared By
System Resources Corporation
128 Wheeler Road
Burlington MA 01803

Table of Contents

Symbols and Abbreviations

Δ	Delta or change
AAR	Airport Arrival or Acceptance Rate
AATT	Advanced Air Transportation Technologies
ACES	Adaptation Controlled Environment System
AND	Approximate Network Delays
AOC	Airline Operational Control Center
AOPA	Aircraft Owners and Pilots Association
API	Application Programming Interface
ASD	Aircraft Situation Display
ATC	Air Traffic Control
ATCSCC	Air Traffic Control System Command Center
ATM	Air Traffic Management
ATS	Air Traffic Service
AWSIM	Air Warfare Simulation
CDM	Collaborative Decision Making
CHATS	Cooperative Human-Adaptive Traffic Simulation
CNS	Communications, Navigation and Surveillance
CONUS	Continental United States
CUBE	Command and Control Unified Battlespace Environment
DDM	Data Distribution Management
DID	Digital Image Design
DMSO	Defense Modeling and Simulation Office
DoD	Department of Defense
DSI	Defense Simulation Internet
ETMS	Enhanced Traffic Management System
FAA	Federal Aviation Administration
FEDEX	Federation Executive
FOM	Federation Object Model
FSM	Flight Schedule Monitor
GUI	Graphical User Interface
HLA	High-Level Architecture
IEEE	Institute of Electrical and Electronic Engineers
IFR	Instrument Flight Rules
ISP	Internet Service Provider
JFAN	Joint FAA/Army/NASA
JSIMS	Joint Simulation System
LAN	Local-Area Network
LMI	Logistics Management Institute
M&S	Modeling and Simulation
MASC	Modeling and Simulation Center
MIT	Massachusetts Institute of Technology
NARIM	National Airspace Resource Investment Model
NAS	National Airspace System

NASA	National Aeronautics & Space Administration
NASM	National Air and Space (Warfare) Model
nmi	nautical miles
NSC	National Simulation Capability
ODMT	Object Model Development Tool
OMT	Object Model Template
OPGEN	Optimized Trajectory Generator
ORD	Operational Requirements Document
RTCA	Radio Technical Commission for Aeronautics
RTA	Required Time of Arrival
RTI	Runtime Interface
SIPRNET	Secret Internet Protocol Routing Network
SOM	Simulation Object Model
SRC	System Resources Corporation
SUA	Special Use Airspace
TCP/IP	Transmission Control Protocol/Internet Protocol
TFM	Traffic Flow Management
TIN	Technical Interoperability Network
TM	Traffic Management
TMU	Traffic Management Unit
TRACON	Terminal Radar Approach Control facility
TSO	Time Stamp Message Order
UDP	User Datagram Protocol
VFR	Visual Flight Rules
WAN	Wide Area Network

Executive Summary

The aviation community in the United States and increasingly in the world has come to a consensus that free flight is a desirable goal, and policies are being initiated to move in that direction. Free flight requires an evolution away from ground-based surveillance and control toward future operational concepts with fewer restrictions on airline and pilot decisions.

However a large question remains: will free flight work as advertised? In a completely changed air traffic system governed by free flight, with stakeholders having differing objectives which are achieved through interaction (cooperation, competition or a mixture of both), will the promised benefits be realized? In what regions of airspace and under what conditions can free flight be allowed without compromising safety?

To better understand these issues and unforeseen problems that might occur in a free flight environment, NASA has proposed a Cooperative Human-Adaptive Traffic Simulation (CHATS). CHATS will be specifically oriented toward simulating strategic decision-making by airspace users and by the service provider's traffic management personnel, within the context of different airspace and rules assumptions. It will use human teams to represent these interests and make decisions, and will rely on computer modeling and simulation to calculate the impacts of these decisions.

The following questions are raised in planning CHATS development:

- Is such a simulation feasible?
- If feasible, how should it be designed?
- If not feasible, what are the alternatives?
- What is the best approach to achieve NASA's objective of better understanding the free flight domain?

NASA issued Task Order 1 under the Advanced Air Transportation Technologies (AATT) program to address these issues.

The first objective of Task Order 1 was to examine the feasibility of implementing the CHATS capabilities as described above. A feasible simulation design has in fact been created. The second objective was to propose a CHATS development plan. The design and the development plan are documented in this report.

An operational concept and a functional design have been created for CHATS. If implemented, CHATS will include the following elements:

<u>Decision makers.</u> These will include personnel representing airspace users, traffic management, airspace and rules specialists, and a team that will create policies governing the simulations. Each team will operate in its appropriate area of influence. Negotiation between air carriers and

traffic management players, and among air carriers, will be a major activity during a simulation run.

Computer simulations. Simulations will be use to evaluate the impact of the decisions made and the events introduced into the scenarios.

Events. Scenarios will be designed and run which introduce external problems, such as bad weather, airport capacity reductions, special use airspace (SUA) activation, and out-of-service aircraft. CHATS will handle initial flight schedules, flight plans, and modifications to flight plans. Air carrier players will be able to modify their fleet deployment based on the course of events in the simulation, and traffic management players will be able to manage airspace and traffic.

More fundamentally, air carriers will be able to propose wholly revised schedules and show their implications. Also the simulation policy team can introduce new airports or airport expansions.

Support. CHATS is designed as a distributed system, with a central operations complex and user, traffic management, and potentially other decision makers using remote workstations at their normal places of business. This will minimize travel time for these personnel. The attention they give to the simulation could be on an as-needed basis.

A development plan is presented under which CHATS can be built and run for a little over $2 million, over a period of three years. During the first year a rapid prototype will be developed to create a functional and user-oriented system. During the second year, the prototype will be utilized in an operational mode to examine interactions and communications between traffic management and airline operational control centers (AOCs), while the full system is being built through an expansion of the prototype capabilities. During the third year, the full system will be utilized in an operational mode. The $2 million life cycle cost is evenly divided between development and operations.

CHATS is essential to explore user competitive and cooperative behavior in a free-flight environment, areas not being addressed elsewhere within the aviation community. Its development and use will permit NASA to become a leader in free flight concepts and assessments. A first-year program which would create a ready-to-run prototype system would cost $550K.

1.0 Introduction

1.1 Background

Free flight has been defined by the RTCA Task Force on Free Flight Implementation (RTCA, 1995) as "… a safe and efficient flight operating capability under instrument flight rules in which the operators have the freedom to select their path and speed in real time. Air traffic restrictions are only imposed to ensure separation, to preclude exceeding airport capacity, to prevent unauthorized flight through Special Use Airspace (SUA), and to ensure safety of flight. Restrictions are limited in extent and duration to correct the identified problem."

In the years since the RTCA report, the term "free flight" has been expanded to apply to a variety of proposed air traffic operational concepts. In the context of this report and project, we will define it as implying future operational concepts with fewer restrictions on airspace user decisions than in today's air traffic control environment. Hereafter, the term "user" or "airspace user" will be used to denote stakeholders who own and operate aircraft, including air carriers, air taxi, general aviation, and military. The "service provider" is the entity which manages air traffic control services, which in the United States is the Federal Aviation Administration (FAA).

Free flight is assumed to have the following attributes or characteristics:

- Satellite-based communication, navigation and surveillance (CNS)
- An air traffic management (ATM) system which unifies air traffic control (ATC) and traffic flow management (TFM)
- Cockpit flight information/management systems to support pilots
- Comprehensive decision support systems for controllers
- Collaborative decision making between airspace users and the service provider
- Air traffic control, to the maximum extent feasible, manages by exception rather than by direction

The following goals have been accepted by consensus of the aviation community as part of the free flight concept (RTCA, 1995):

- To encourage owners to equip aircraft with the necessary technical aids by providing user benefits
- To enable maximal decision-making by pilots and air carrier operational control centers (AOCs)
- To change the role of the service provider from managing by direction to managing by exception

The following benefits have been claimed from implementing free flight (RTCA, 1995):

- It will allow the service provider to continue adequate air traffic service in the face of a rapid, continuing increase in air traffic demand
- It will reduce user costs due to improved scheduling and routing efficiencies
- It will increase air carrier flexibility and enable them to better optimize their operations
- It will produce time savings to passengers and shippers
- It will improve service provider efficiency of operation

1.2 Problem Statement

The aviation community in the United States and increasingly in the world has come to a consensus that free flight is a desirable goal, and policies are being initiated to move from the current air traffic control framework to the new one described above. In particular, the Federal Aviation Administration (FAA) has made free flight the foundation of their air traffic concept of operations to guide future system planning (see for example FAA, 1997(2)). A large question, however, remains: will free flight work as advertised? In a completely changed air traffic system governed by free flight, with stakeholders having differing objectives which are achieved through interaction (cooperation, competition or a mixture of both), will the promised benefits be realized? In what regions of airspace and under what conditions can free flight be allowed without compromising safety?

To better understand the issues and unforeseen problems that might occur in a free flight environment, NASA has proposed a Cooperative Human-Adaptive Traffic Simulation (CHATS). CHATS will be specifically oriented toward strategic decision-making by users and by the service provider's traffic management personnel, within the context of different airspace and rules assumptions. It will use human teams to represent these interests and make decisions, and will rely on computer modeling and simulation to calculate the impacts of those decisions.

Objectives

The following objectives have been defined for CHATS:

<u>Develop a simulation capability which focuses on user and service provider strategic decision-making in the free flight environment.</u>

The simulation will emphasize strategic as opposed to tactical decisions. Examples of strategic decisions are planning and re-planning flight schedules for a fleet, and type and duration of traffic restrictions. The time horizon for strategic decisions ranges from hours to days to (potentially) years. Examples of tactical situations are collision alerts and avoidance, and changes in arrival sequencing when an aircraft is near the airport. The time horizon for tactical decisions is on the order of minutes.

<u>Assess new roles and strategies for traffic management.</u>

An example of a new strategy for traffic management is to replace the objective of optimal system-wide traffic flow by allowing each air carrier to optimize its own operations, with traffic management maintaining controller effectiveness.

Test competitive and cooperative strategies.

The simulation will have differing ground rules. During some runs each user group will execute its own strategy without any communication with others. During other runs, users will share information as they choose. There could be many variations showing different levels of competition and cooperation.

Determine impacts of these strategies upon stakeholders (users and providers).

The simulation will be designed with metrics to measure results such as flight delays, missed connections, airline operating cost impacts, and sector loading.

Find out, during planning and execution of CHATS, the issues important to stakeholders.

Involvement of stakeholders while planning CHATS will allow identification and prioritization of issues and outcomes such as workload, staffing, delays and their associated costs, and fleet utilization. Additional insights will come out of scenario execution.

Permit users and providers to invent and evaluate new strategies.

If full free flight were approved for use today, do the users and service providers have strategies to follow to gain the advantages promised? CHATS will help users and service providers to develop and evaluate such strategies.

Permit providers to study effects of new airspace structures and rules.

From the government perspective, CHATS simulations could test the effect of the abolition of fixed routes as compared with today's web of fixed routes, or other less radical changes in route structure.

Development Issues

The following questions are raised in planning CHATS development:

- Is such a simulation feasible?
- If feasible, how should it be designed?
- If not feasible, what are the alternatives?
- What is the best approach to achieve NASA's objective of better understanding the free flight domain?

NASA issued Task Order 1 under the AATT program to address these issues.

1.3 Approach

The first objective of Task Order 1 was to examine the feasibility of implementing the CHATS capabilities as described above. A feasible simulation design has in fact been created. The second objective was to propose a CHATS development plan. The design and the development plan are documented in this report.

The Task Order 1 work was divided into two phases. Phase I assessed the technical feasibility of meeting the CHATS desired capabilities as determined from the NASA Statement of Work. The conclusion (SRC, 1998) was that most of the capabilities could be created in a technical or physical sense, since the risks were low to moderate. The following capabilities had high risk and therefore will be more difficult to create:

- Combine computer-assisted cooperative work methodology and new or existing simulations
- Estimate system stakeholder tendencies to compete and cooperate
- Determine if stakeholder competitive actions could jeopardize concept feasibility
- Determine levels, types, and timeliness of information sharing that is necessary in the conflict detection and resolution process

Based on these results, System Resources Corporation and the NASA Task Manager decided in Phase II to create a feasible functional design to accomplish the first three of the above capabilities, while leaving involvement in the conflict detection and resolution process and related subjects for future work. This decision is due to NASA's interest in simulating users' and service providers' strategic behavior, research which is not being conducted elsewhere within the AATT project.

Appendix A shows which of the desired CHATS capabilities from Phase I will be satisfied by the CHATS functional design described in this report. An explanation is given for those capabilities that are not satisfied. In general the reason was in order to develop a feasible simulation and one that could be implemented within reasonable time and cost limits.

2.0 CHATS Operational Concept

The operational concept is discussed with respect to potential human actors or players within the simulation, a rationale for choosing players to represent particular functions, how the simulations will be conducted, and some questions that the evaluations should be designed to answer.

Players

The corporate entities involved in operating flights, and supplying players representing their interests, can be divided into air carriers, air taxi, general aviation, military, and the service provider (FAA). The following describes the types of personnel involved for each entity.

Air Carriers. Air carriers are defined by the FAA for statistical purposes as for-hire carriers operating aircraft with more than 9 passenger seats or 7,500 pounds payload capacity. These include integrated carriers of passengers and freight, both scheduled and charter, and air freight specialists including package carriers. Virtually all air carriers have in common the need to plan and schedule fleets of aircraft. The following types of personnel are involved in air carrier operations. Personnel are listed in order of tactical to strategic in the duties of the position.

1. Pilots/flight crews
2. Station Managers
3. Ramp Control personnel
4. Dispatchers
5. ATC coordinators
6. Crisis Management personnel
7. Scheduling personnel
    - Gate management
    - Flight crews
    - Aircraft maintenance
    - Flight schedules
8. Marketing/Fleet Mix decision makers

Air Taxi. Air taxi are defined as for-hire carriers operating aircraft with 9 or fewer passenger seats and less than 7,500 pounds of payload capacity. There are two general classes of air taxi operators. One class consists of small commuter carriers which run scheduled services usually code-shared with a major airline. These require fleet management and will have the same types of personnel listed above or will use a major airline's planning resources. The other class consists of independent for-hire aircraft. These operate in a similar manner to general aviation for the purposes of the CHATS concept.

General Aviation. These are defined as all civilian aircraft not for hire. They include corporate aircraft for executives, and individually owned and piloted aircraft. General aviation does not require the kind of fleet management practiced by air carriers. Personnel include:

1. Pilots
2. Aircraft owners

Military. The following is a partial list of military personnel roles in aviation:

1. Pilots
2. TRACON controllers
3. Fleet exercise planners (including SUA planning)
4. AOC or equivalent personnel
5. Traffic management personnel

Service Provider. With the exception of weather forecasters and some contract tower controllers, these are all FAA personnel.

1. Tower controllers
2. TRACON controllers
3. ARTCC controllers
4. Weather forecasters
5. Traffic management personnel in Towers and TRACONs
6. Traffic Management Unit (TMU) personnel in Centers
7. System Command Center personnel
8. Air Traffic Operations Requirements personnel
9. Air Traffic Airspace and Rules personnel

Use of Players in the Simulation Environment

User Teams

It is proposed that user teams represent the following interests:

Airlines. A small number of airline (including commuter) teams is proposed for CHATS, each of which will include personnel relevant to a given simulation. These normally would include the following:

- Dispatchers (flight re-planning function)
- Scheduling personnel
- Marketing/Fleet Mix decision makers

who most represent strategic as opposed to tactical decisionmakers. The remaining players listed above under air carriers will have their activities represented by a computer simulation or as part of the scenarios employed.

It will be beneficial to have at least two major competitive airlines represented by teams, to explore their competing and cooperative behavior in future air traffic environments.

Air Cargo Operators. These have distinctive operating procedures from passenger-carrying airlines; for example, night operations and a very focused hub organization. For this reason it would be beneficial to have at least one air cargo operator team involved. The personnel will be similar to those representing airlines.

General Aviation. A general aviation team would represent the interests of individual non-scheduled flights. The team personnel typically would include a corporate owner, a general aviation pilot taking the role of flight planning and re-planning, and perhaps a representative of the Aircraft Owners and Pilots Association (AOPA).

Military. A military team would represent the interests of military flights which use FAA-controlled airspace during peacetime or to support overseas operations. Although some of these flights are coordinated by an operations center, for example within the Air Mobility Command, most are planned and flown as individual non-scheduled operations similar to those of general aviation. The team personnel typically would include planning specialists from a military operations center, and a pilot performing flight planning and re-planning for individual operations.

Service Provider Teams

The service provider will be represented by two teams: a traffic management team, and an airspace and rules team. The traffic management team will include traffic managers and ATC operations requirements personnel, as relevant to a given simulation. The airspace and rules team will be represented by FAA specialists in these areas. The remaining players listed above under the service provider will have their activities represented by a computer simulation.

Concept Definition and Experiment Design Team

This team will control the design, conduct, and evaluation of the simulations. It will consist of a simulation policy team, and the airspace and rules team defined above. The policy team will be led by NASA and FAA officials, and will define concepts to be evaluated and establish simulation ground rules.

Figure 2-1 summarizes the teams as described above. It is anticipated that these teams will be geographically distributed and therefore need to be connected by a wide-area network. A simulation operator will run the simulations and will have a workstation controlling computer simulations as described in Section 3 and connected by a local-area network.

Simulation Conduct

The following is a generic description of anticipated simulation conduct. A more detailed description and examples of common scenarios are given in Section 3.

The concept definition and experiment design team will consist of the coordinated activities of the simulation policy team and the airspace and rules team. The simulation policy team will first define concepts to be evaluated, for example finding the domain of feasibility for a certain free flight concept assuming double today's air traffic volume. It will then establish simulation ground rules, for example whether to allow inter-airline communication and coordination, and defining the speed of onset and extent of a weather front. It will develop a detailed experiment design to carry out a series of simulation runs.

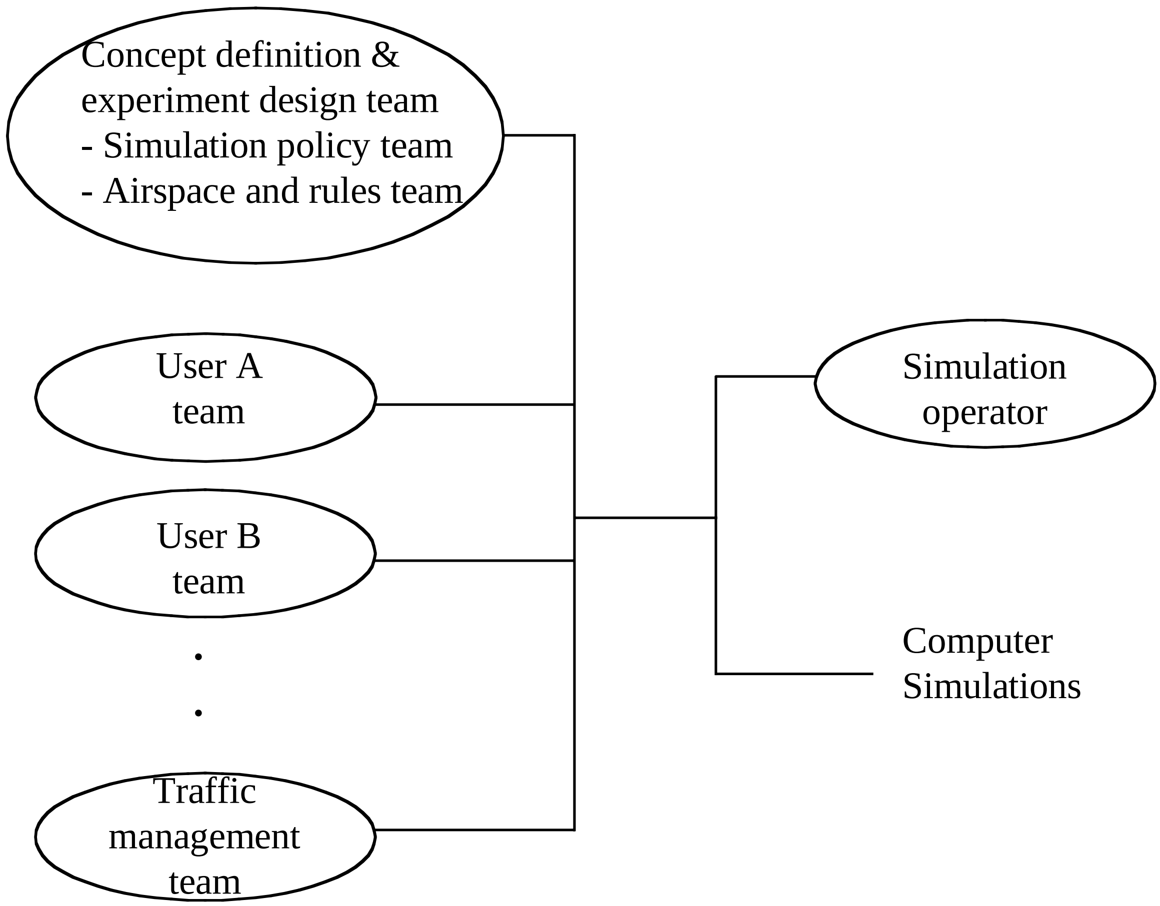


Figure 2-1. CHATS Teams

The airspace and rules team will establish a set of airspace and rules assumptions. Route assumptions may allow full freedom of user-preferred routes, all fixed routes, or something in between. The team will define the roles and responsibilities of the traffic management team. It will impose rules such as requiring ground delay programs to take the place of air delays.

User teams will establish their initial objectives. A key part of the CHATS philosophy is that these objectives may change as user teams gain experience with the future concepts that are being tested. As the objectives of different users and their strategies change, there may be interesting and unpredicted effects on the outcomes of the simulation runs. This is the "cooperative human-adaptive" part of this project.

User teams with scheduled operations will supply initial flight schedules for a day or a week, as the basis of a simulation run, in accordance with the future air traffic environment set by the simulation policy team. For example, if the overall air traffic volume increased by 50% over the current volume, the team may be directed to submit a schedule with 50% increased operations over today. The team may also assume service on new flight segments or to new airports as compared with today's schedule. Decision support tools will help in preparing these schedules, and will also be used to support re-planning activities.

The traffic management team will set traffic management objectives in coordination with the airspace and rules team. For example, the team may try to maintain system efficiency as in today's operational concept, or they may remove this objective entirely and allow the users to maximize their own individual efficiencies. These objectives may also evolve and change in ways initiated by the learning experience of the team, for the same reason as stated for the user teams. The team will first receive the user-requested schedules and combine these with other predetermined schedules and a realistic mix of general aviation and military flights. They will do a "look-ahead" trial simulation run, determine problems and negotiate schedule and flight plan changes with the airlines to avoid sector overloads and to adjust to airport capacities. After this initial adjustment, the traffic simulation will begin.

The simulation will normally run in fast time to conserve time for the players. It can be paused as appropriate to examine problem "events" and decision points for the teams. A classification of such events needs to be determined and would include such items as weather problems and facility saturation. Alternatively, the simulation may be run in real time without pauses. This would force the airline players to react to developing problems in a realistic response manner. This makes the exercise closer to the real world.

All procedures will be under the control of the simulation operator. The operator will run simulations according to the ground rules, coordinate teams, and collect data in accordance with the experiment design. Adjustments may be made in schedules and flight plans at the pause times to resolve unacceptable situations from the perspective of traffic management or to gain advantage for an airline. Communication systems will allow traffic management – airline coordination and, if allowed under simulation ground rules, inter-airline coordination. Coordination methods from the Collaborative Decision Making (CDM) program, both currently employed and as proposed, will be part of the simulation.

During the simulation run interesting problems may be introduced, following the simulation ground rules and as controlled by the simulation operator. These may be introduced manually or as a result of random processes. These could include current weather requiring route changes, airport restrictions or closures, establishment of Special Use Airspace (SUA) restrictions, airborne and ATC equipment failures, etc. These will be communicated to the human players and to the computer simulation, leading to the appropriate changes in the simulation run.

Evaluation of Results

A number of questions will be posed that the simulation will try to answer, and metrics will be developed to help compare the results of different simulation runs against each other. The evaluation will be from the perspectives of the system, the service provider, and the users.

Examples of such questions are the following:

- How will limited airspace and airport resources be allocated in the future, and under increased demand? Are these allocations satisfactory from the point of view of the different stakeholders?

- Is the allocation process stable, including the effects of negotiation, competition, and disturbances, or does it break down?
- How well do future concepts respond to unusual events such as bad weather, a closed runway, or equipment failures?
- Will the airlines get information they need to manage their banks of flights and maintain schedule integrity?
- What kinds of criteria do the teams, representing different stakeholders, use to make decisions?

3.0 CHATS Functional Design

The CHATS functional design is described as follows. First, dataflows are described to show the major modes of communication among the human teams and the models and simulations. Second, general functional processes are discussed, along with communications and operations within the distributed simulation. Third, specific problems and scenarios are described which the simulation is designed to address. Fourth, result metrics are defined and discussed.

3.1 Dataflows

Six data flow diagrams are shown, organized in a hierarchical manner. In these diagrams the following symbols are used:

- An elliptical box represents a team
- A rectangular box represents a process or model exercise
- A hexagonal box indicates a process external to the principal subject of the diagram
- The circular file storage symbol represents a database or dataset
- A rounded rectangle represents an information display

All major information flows are shown, but in order to avoid clutter not all the external inputs and influences from the player teams are shown. These are shown in the functional process diagrams in the next section.

Figure 3-1 shows high-level data flow. The major functions shown here, namely Traffic Generation and Tracking Tools, Airport and Terminal Area Queuing Model, Traffic Management Team, and User (Air Carrier) Teams are described in succeeding charts.

The concept definition and experiment design team generates an operations concept and ground rules which are input to traffic management, the user teams, and the traffic generation and tracking tools. The simulation operator generates one or more scripts to be followed in the conduct of the simulation which conform to these defined assumptions. Part of the script is the creation of external events such as weather, winds, and SUA use, either by manual parameter setting or by use of a random process. These are also input to the tools.

Flight schedules are provided by air carriers if they are represented in this simulation run as players, otherwise archives from the Enhanced Traffic Management System (ETMS) can be used to fill out a complete schedule. Schedules and user objectives are also input to the tools.

During a simulation run, the tools provide system status both to traffic management and to user teams in a manner similar to what is done in today's CDM environment using the Aircraft Situation Display (ASD) from ETMS and the Flight Schedule Monitor (FSM) display. Based on status, the teams enter changes which will be described in later diagrams.

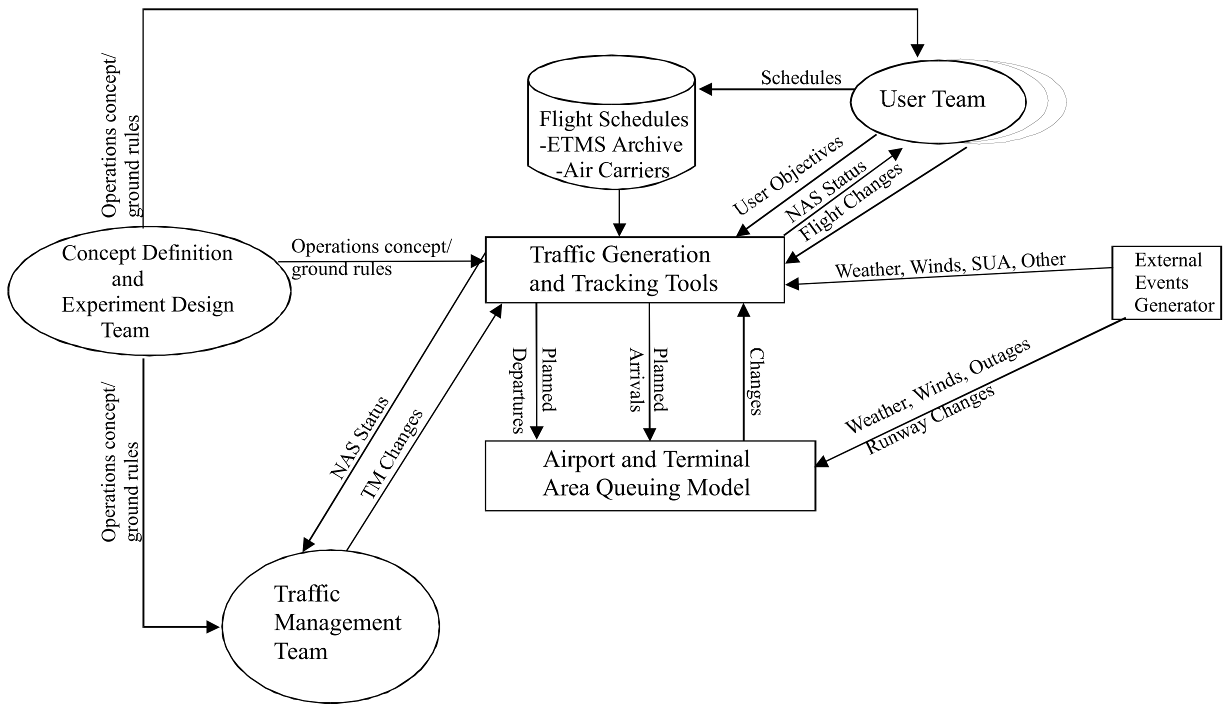


Figure 3-1. High-Level Data Flow

The Airport and Terminal Area Queuing Model represents the constraints caused by airport demand-capacity imbalances. Weather, winds and airport configuration are the external drivers of airport capacity. The model accepts as inputs planned arrivals and planned departures at capacity-constrained airports, and sends changes back to the tools.

In the following discussion, it should be noted that we are recommending building CHATS using to a great extent existing models and tools as described herein.

Figure 3-2 shows the Traffic Generation and Tracking Tools. The three principal tools are the Optimized Trajectory Generator (OPGEN), the Total Traffic Tool, and the Find Crossings Tool. These are the three parts of the National Airspace Resource Investment Model (NARIM). The model was developed for the FAA and has been used by the FAA and the NASA Langley Research Center. Some modifications will have to be made to these tools to adapt them to future air traffic operational concepts.

The OPGEN takes flight schedules and calculates optimized planned trajectories with inputs of SUA, aircraft performance, weather and winds. Operational concept assumptions and user objectives will also influence how the model performs its calculations.

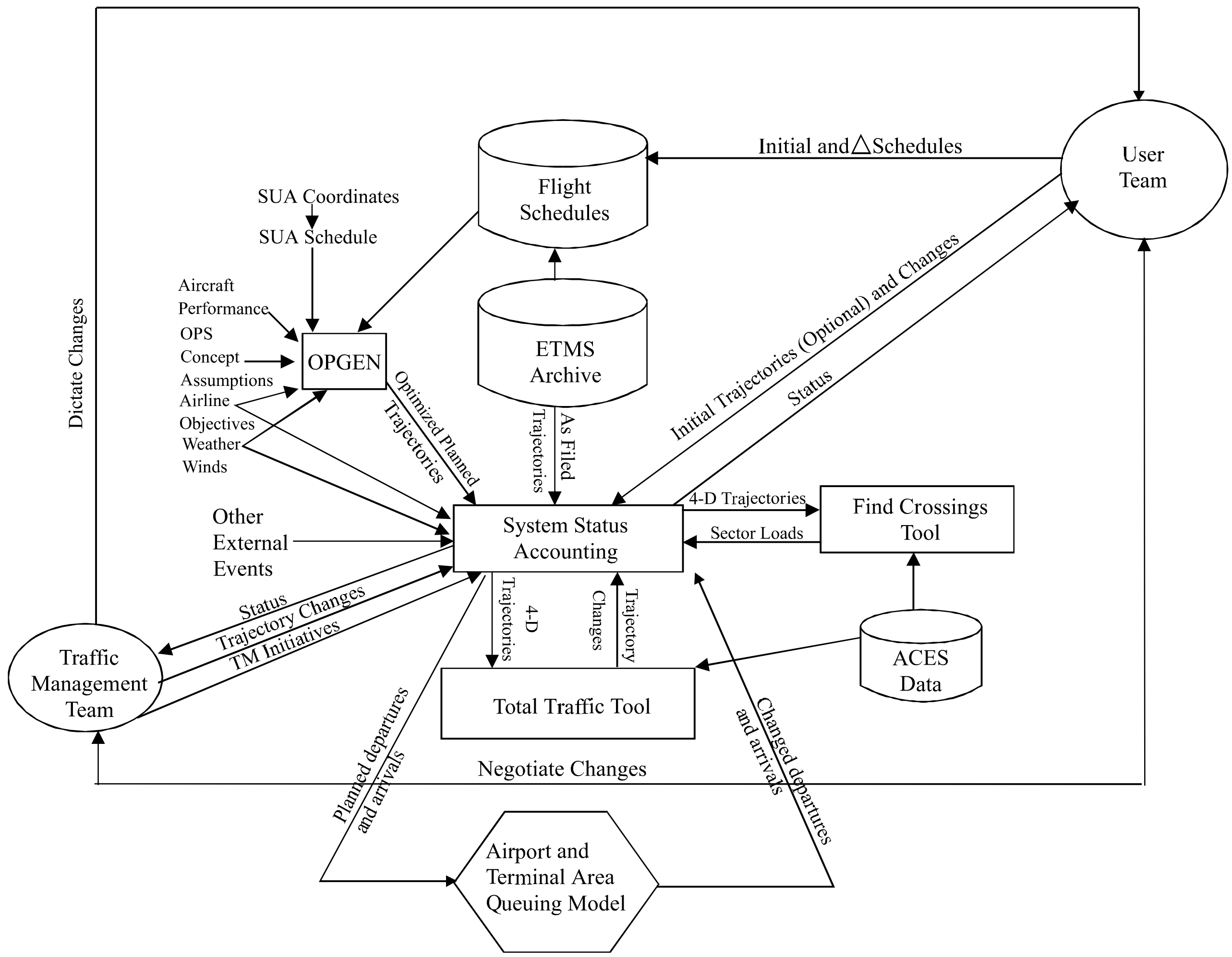


Figure 3-2. Traffic Generation and Tracking Tools

In the real world, the air carrier dispatchers and pilots prepare and file their own flight plans. OPGEN provides realistic trajectories for scheduled flights not represented by air carrier teams. Air carrier teams will have the option of directly submitting trajectories based on their own flight plans, or allowing the model to calculate initial trajectories and submit modifications as necessary.

A system status accounting model starts with optimized planned trajectories and adds the trajectories based on filed flight plans for unscheduled flights, including representation of general aviation and military flights, from an ETMS archive. The result is a representation of all flights within the continental U.S. The status model will provide the current state of the National Airspace System to all teams, including flights, weather, airport conditions, etc.

As weather and winds change, OPGEN can be rerun on a regular basis to alter the optimized trajectories from that time forward. Weather has two components in the simulation: forecasts (which can be wrong), and current weather. Other external events such as aircraft problems will change the status. Stochastic disturbances of the trajectories can also be introduced.

The 4-D trajectories are used by the Total Traffic Tool to detect aircraft conflicts. An extension to this tool will estimate delays which result from simulated controller actions to avoid the conflicts. These are fed back to the system status as trajectory changes. The trajectories are also used by the Find Crossings tool to predict sector loads, which become part of the system status. Both of these tools require inputs on the airspace structure from the Adaptation Controlled Environment System (ACES), namely fixes, sectors, and airways. ACES data can be obtained from the ETMS database.

The system status is transmitted to traffic management and to the user teams, and these teams will make decisions to cause changes to occur to meet their objectives. The air carrier teams may plan changes to their schedules, generally after negotiation with traffic management. In addition air carriers may input direct trajectory changes based on changed flight plans to meet arrival constraints. If air carriers are not represented as players, traffic management will make changes in flight schedules based on their decision criteria, and they may also input trajectory changes for non-scheduled flights to react to weather and other problems.

Figure 3-3 further describes the Total Traffic Tool, as modified to account for simulated controller action to avoid aircraft conflicts. The tool finds proximity events, that is aircraft coming to close to each other, from the 4-D trajectories. It then makes an assumption that both aircraft involved in the proximity suffer a standard delay because of the controller's action; it does not model the interaction in detail. These delays are fed back to system status as changed trajectories. The changed trajectories will change future proximity events. Note that a single flight has a good probability of suffering multiple delays over its entire route if that route takes it through multiple areas of heavy traffic.

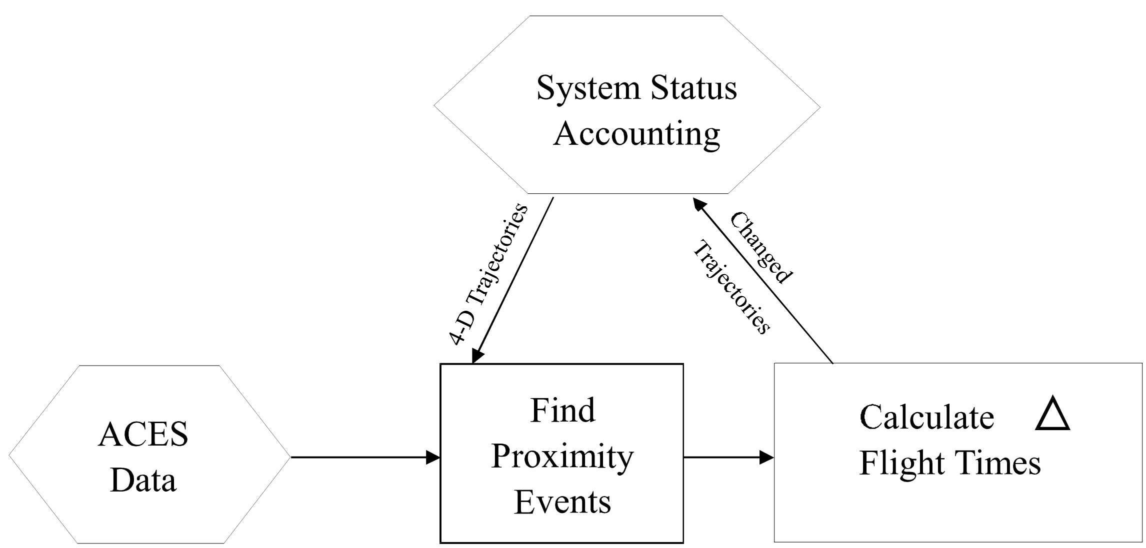


Figure 3-3. Total Traffic Tool

Figure 3-4 describes the queuing model. It has an airport configuration and capacity database which has capacity parameters for each capacity-constrained airport. The capacity depends on the runway configuration and meteorological conditions. The runway configuration depends on the winds. The queuing model takes the current capacity and calculates arrival and departure queues based on planned arrival rates and planned departure rates. These queues cause arrival and departure delays which are fed back to the tools.

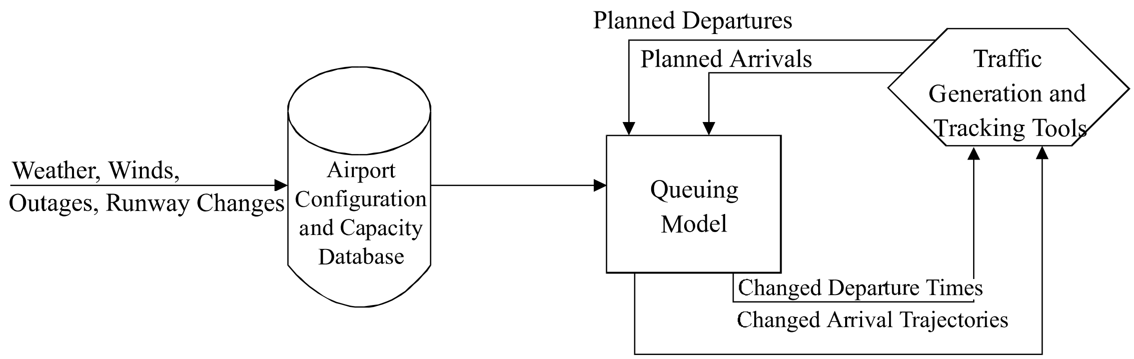


Figure 3-4. Airport and Terminal Area Queuing Model

It is estimated that about 60 airports within the continental U.S. are capacity-constrained and need to be modeled this way. Airport arrival or acceptance rates (AAR) need to be obtained for these airports for each configuration and meteorological condition, and from these a simple capacity model can be developed. Constraints are not anticipated at the other airports because of the expected distribution of traffic.

Two airport queuing models have been created and could be adapted to CHATS: the Approximate Network Delays (AND) model at MIT, and the LMI Capacity Model at the Logistics Management Institute. These are described in SRC, 1998, Appendix B.

Figure 3-5 shows decision making within the Traffic Management Team. In CHATS, as fully operational, it is assumed that the traffic management team operates from the System Command Center. A traffic management operations concept is developed and applied in cooperation with the policy team. The system status is input to ETMS/ASD and the Flight Schedule Monitor (FSM) as displays and decision support tools.

Traffic management prepares trial plans, often with the use of the “look-ahead” feature of the tools, involving ground delay programs, changes due to monitor alert, reroutes due to weather, and CDM measures. As previously discussed these changes may be negotiated with the air carriers, schedules may be dictated, or trajectories may be changed.

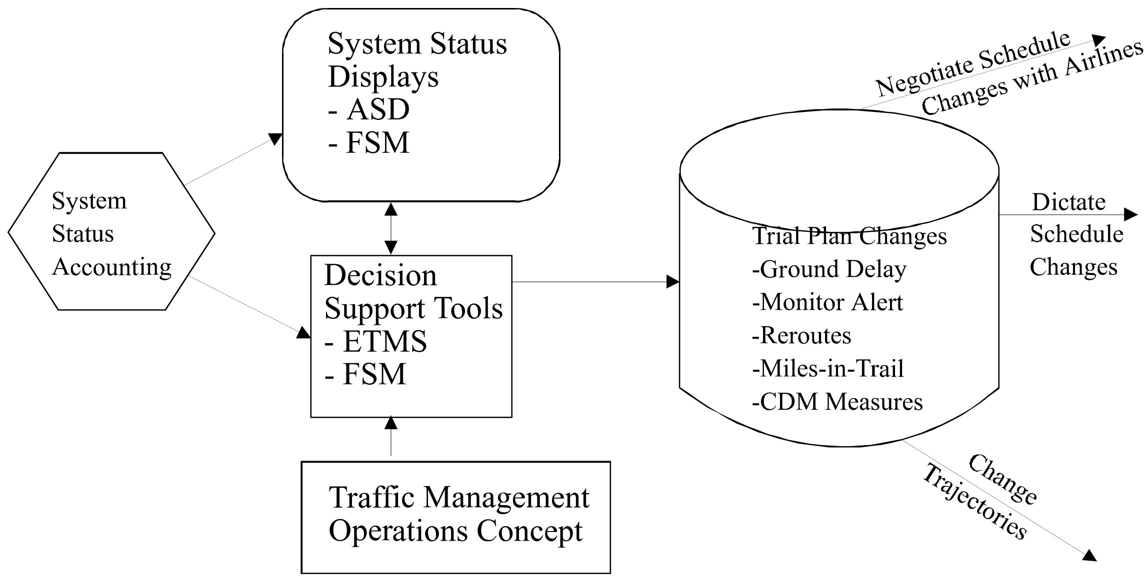


Figure 3-5. Traffic Management Team

Figure 3-6 shows decision making by an air carrier user team. In CHATS, as fully operational, it is assumed that the air carrier teams operate from their AOCs. Team objectives are defined and guide user decisions. The same system status displays as available at traffic management are shown to the air carrier teams, with some information about general aviation, military and competitor air carrier flights removed (this could vary with the ground rules). In addition, an air carrier flight and equipment status database will be created and have an associated data display tool.

Air carrier decisions include a basic weekly schedule, flight/equipment assignments, flight-slot assignments, cancellations, added sections, and earliest departure time estimates. In this simplified simulation environment crew and gate scheduling will not be included. The results of air carrier decisions are negotiated schedule changes with traffic management and, depending on the ground rules, with other user teams; schedule changes input to the flight schedule master database; and trajectory changes from new flight plans.

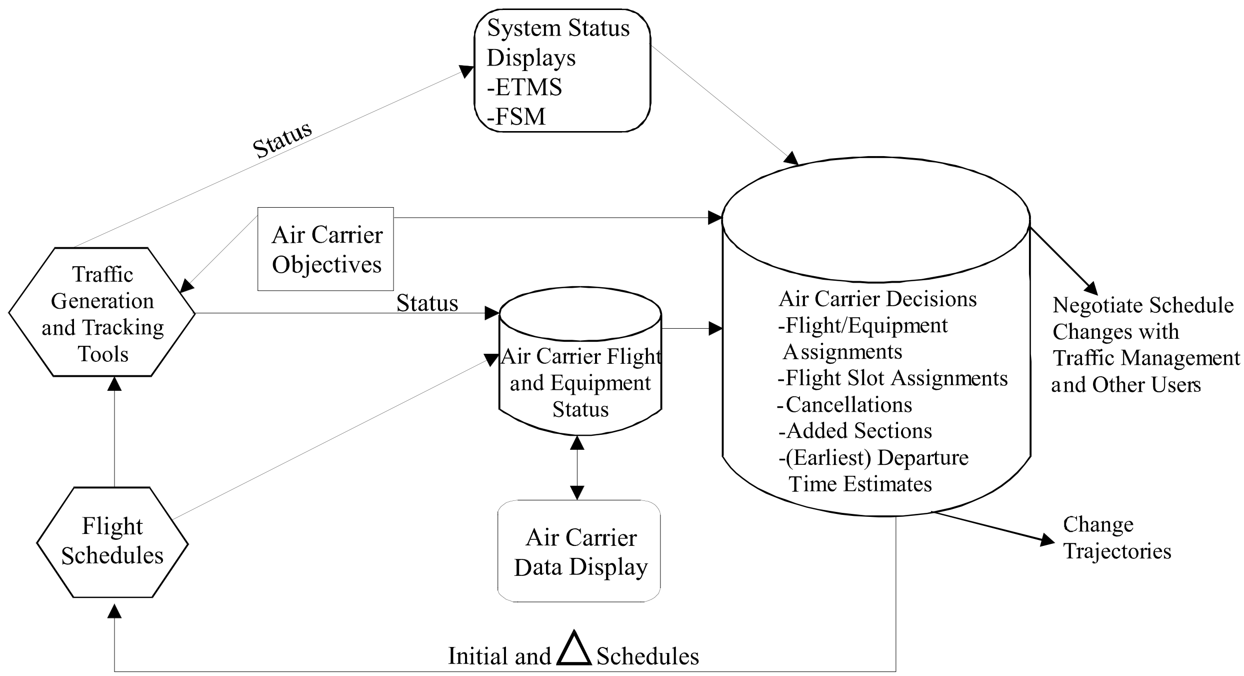


Figure 3-6. Air Carrier User Team

An example of a data display for a passenger airline, called a connection matrix, is shown in Figure 3-7. This diagram and the one following were obtained courtesy of Bill Hall, Operations Research Center, Massachusetts Institute of Technology. It shows at a single airport the arrival flights down the left side in chronological order and the departure flights across the top in order. At the intersection of each row and column is a colored dot indicating whether a connection between the arriving flight and the departing flight is feasible (green), marginal (yellow), or infeasible (red). For example, depending on the airport, a connection may be feasible if there is a 20 minute or more window, marginal if there is a 10-20 minute window, and infeasible if there is less than 10 minutes to make the connection. The chart will be continually updated with current expected arrival and departure times for the flights.

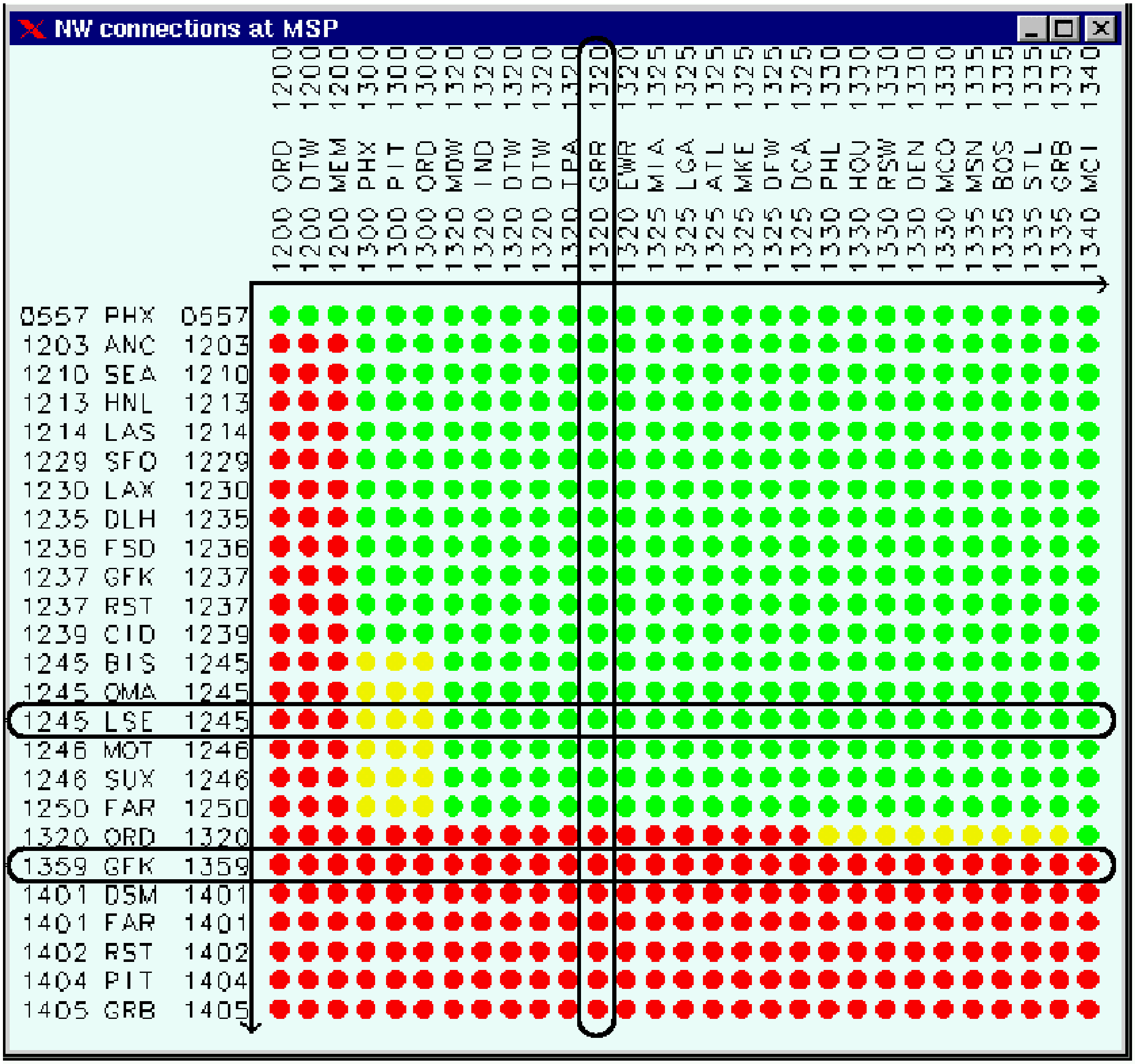


Figure 3-7. Connection Matrix, Example 1

Figure 3-8 shows a more fully developed example. It shows the banking phenomenon where a group of arrival flights are expected to connect with a group of departure flights. In this chart, if weather were to delay the whole bank of arriving flights, more and more dots would start turning yellow and red showing that connections were in jeopardy. The airline could of course delay the departing bank to preserve connectivity and this will cause other problems.

Figure 3-8 also shows two other features. The intensity of each dot is proportional to the number of passengers trying to connect from arriving flight A to departing flight B, thereby allowing a visual estimate of schedule connection integrity. In order to implement this capability in the simulation, passengers need to be included in the airline data base. Also when an arriving flight and a departing flight use the same aircraft, this is indicated by a small black dot at the intersection of the flights.

If slot assignments were displayed along the left side, a useful feature would be to provide the analyst with a “drag-and-drop” capability to see what would happen if arriving flights were to switch slots with each other.

Figure 3-9 shows the Flight Schedule Monitor (FSM) which is currently being used as part of the CDM agreements. The same displays show in the System Command Center and in the AOCs, except that for the AOC the flight identifiers for other than that airline’s flight are filtered out. The FSM would be used in the simulation in the same way as in the real world.

The left panel shows the flight arrival timeline window at an airport. Open arrival slots and cancellations can be viewed. The upper right panel shows color coding for the graph. The middle right panel calls out a particular flight. The lower right panel shows airport capacity vs. demand. When demand rises above the white line showing planned AAR, adjustments must be made. In today’s world, ground delay programs are likely to occur. In the simulated future, new programs could be invented and tried out along with more free-wheeling negotiations.

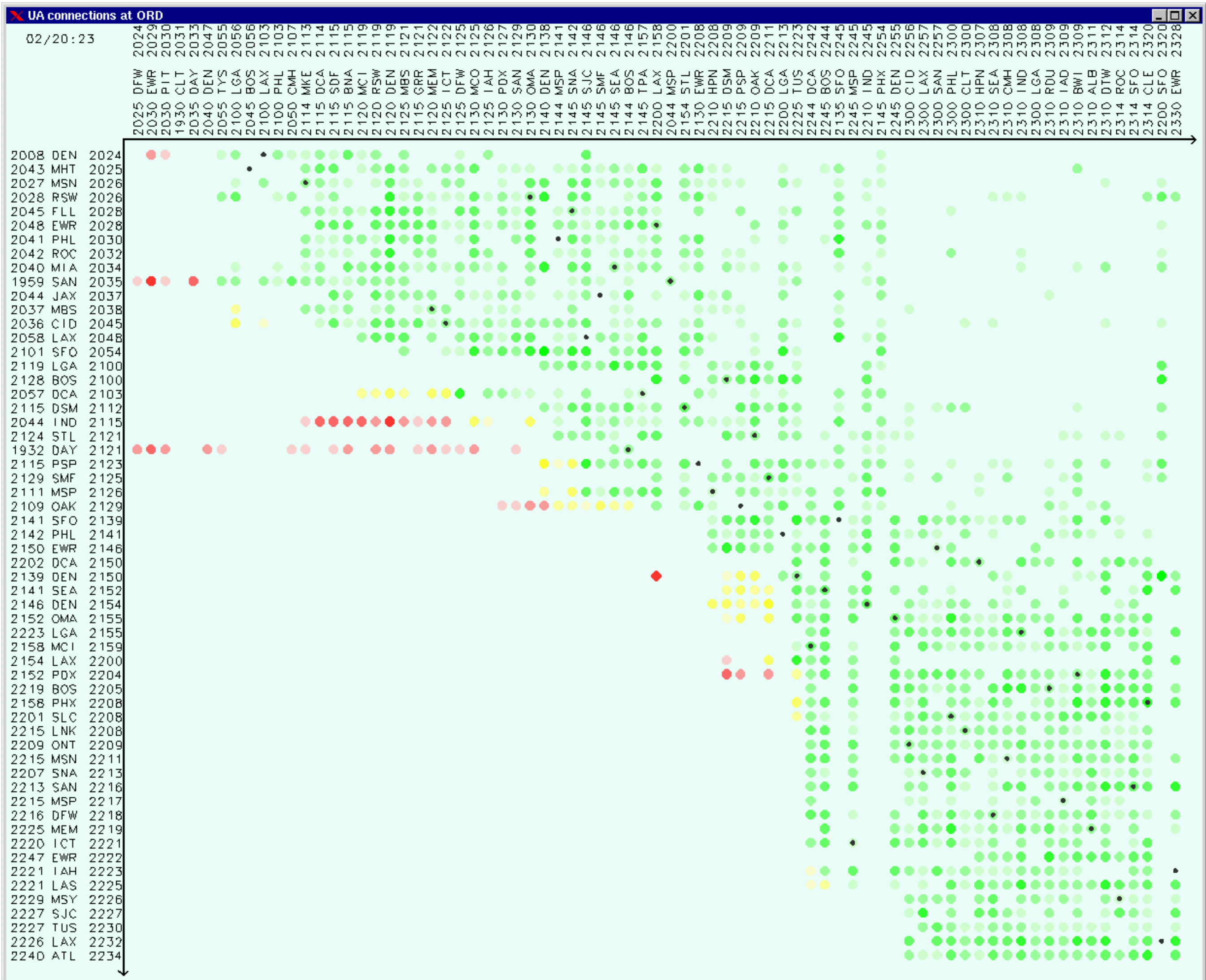


Figure 3-8. Connection Matrix, Example 2

Figure 3-9. Flight Schedule Monitor – Four Windows
(Courtesy of Metron, Inc.)

3.2 Functional, Communications and Operational Procedures

Functional Flows

The functional flow of a generic simulation is shown in a series of ten steps as described and illustrated below. The simulation is oriented toward the decision process of airline user teams.

1. Figure 3-10 shows the first step of the simulation. This involves the Simulation Policy Team defining the assumptions (such as the day of week, year for future scenarios) and external events (such as weather conditions, facility outages). The policy team and the airspace and rules team would jointly define the ground rules for the simulation exercise. Ground rules would include specific implementation options for the concept of operations such as when flight replanning can be performed (for example, can replanning occur throughout the flight or when new information is available can the flight plan be changed significantly?). Other examples would include sectorization and airport configuration.

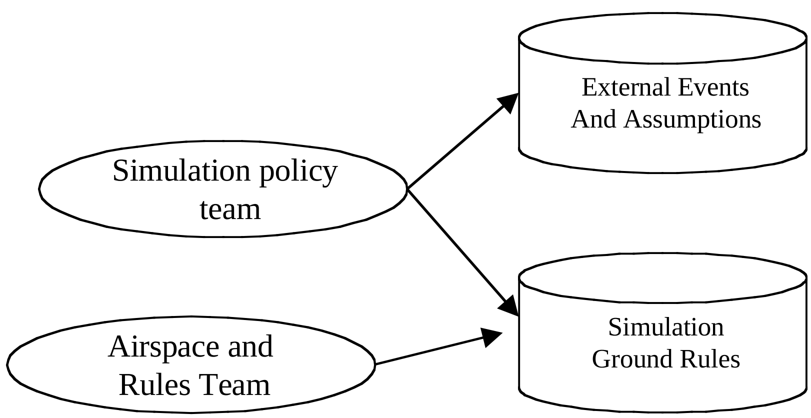


Figure 3-10. Simulation setup

2. Figure 3-11 shows that flight schedules would be derived from ETMS schedule data. These schedules include recent inputs from the airline teams. ACES data would be generated to reflect the simulation ground-rules. The Traffic Management (TM) Team would evaluate these inputs in light of their concept of operations to determine an initial course of action.

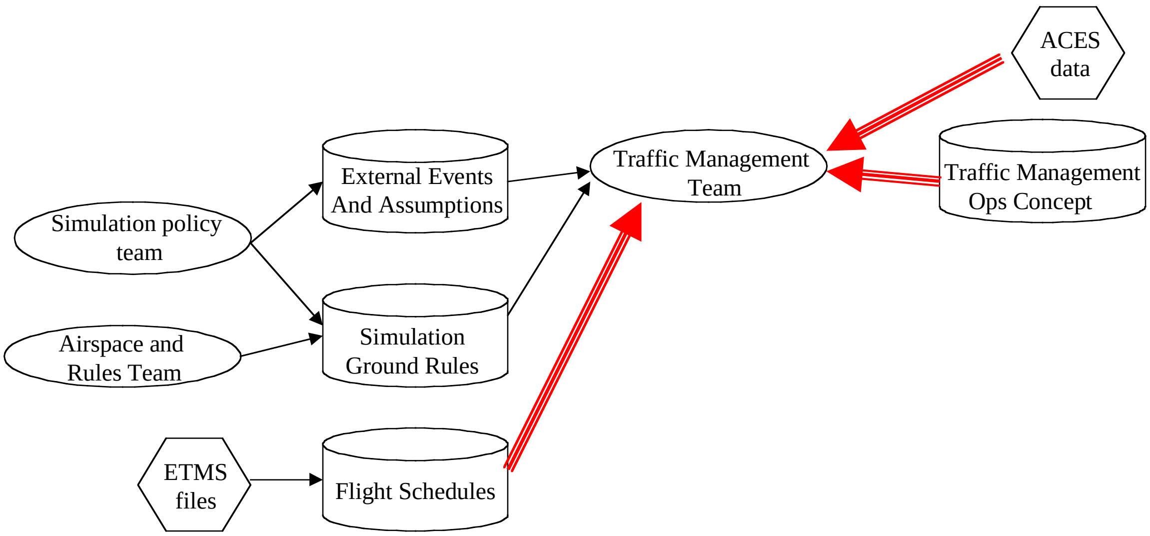


Figure 3-11. TM assesses scenario to determine initial course of action

3. Figure 3-12 shows the TM team taking the course of action by implementing TFM initiatives such as ground delay programs, ground stops and miles-in-trail restrictions. Airline Teams would evaluate these TFM restrictions in light of their individual flight schedules and their flight planning strategy to determine their course of action. Flight schedules are sent to the airline teams, filtered to remove the identity of flights other than each airline's own.

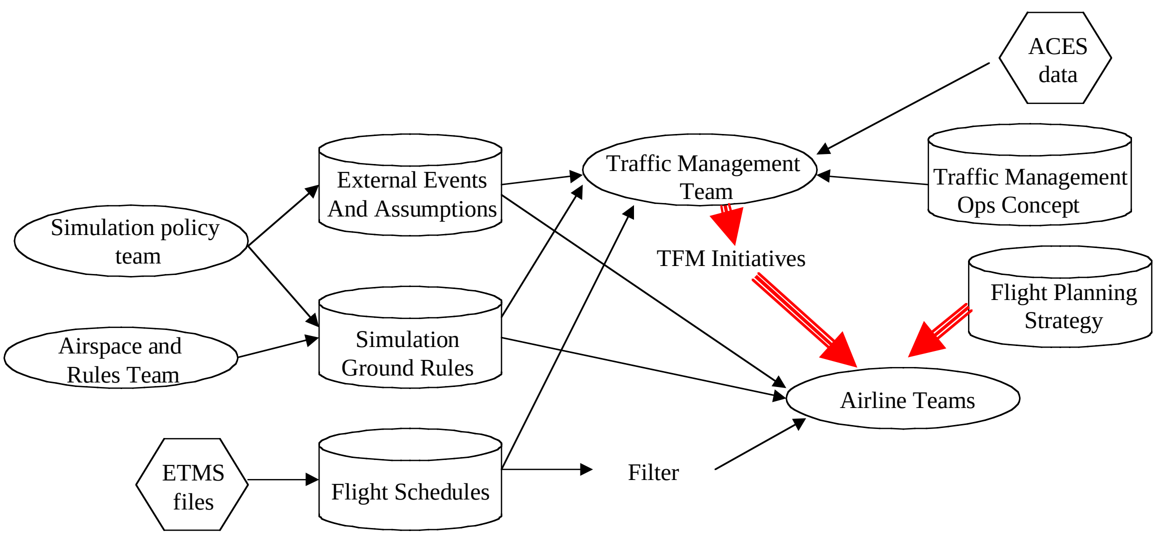


Figure 3-12. TM determines initial course of action

4. Figure 3-13 shows Airline Teams conveying their preliminary course of action, consisting of flight replanning, cancellations and substitutions.

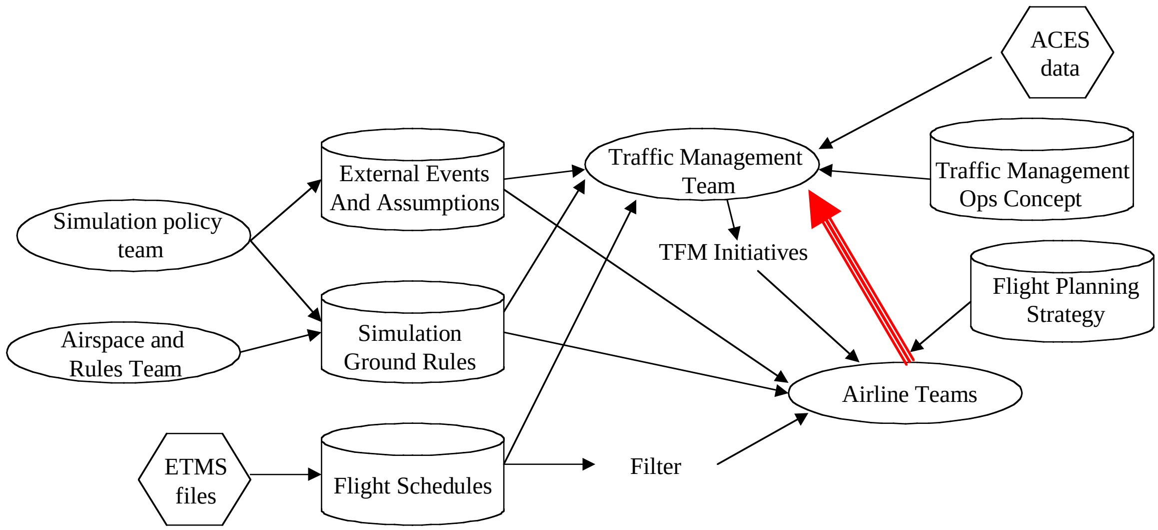


Figure 3-13. Airlines provide feedback on course of action

5. Figure 3-14 shows that based on the modified state of demand, the TM Team re-evaluates the original traffic management initiatives and modifies them as deemed necessary. The Airline Teams evaluate this and change their flight schedules accordingly.

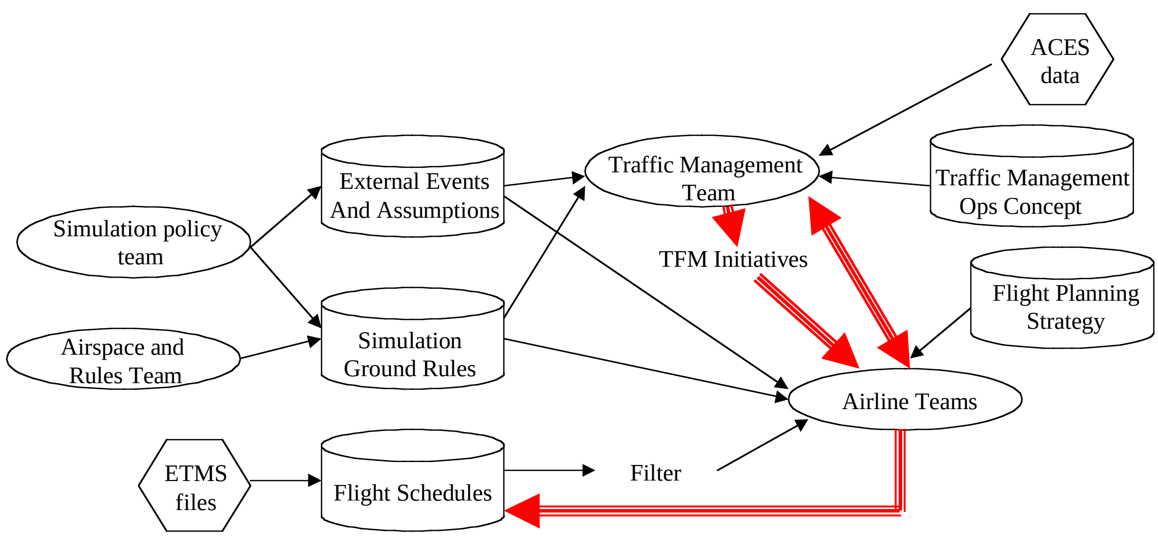


Figure 3-14. TFM provides feedback to airline teams who make further changes

6. In Figure 3-15 the simulation is initialized with the modified flight schedules and initial TFM initiatives. Then the simulation is started.

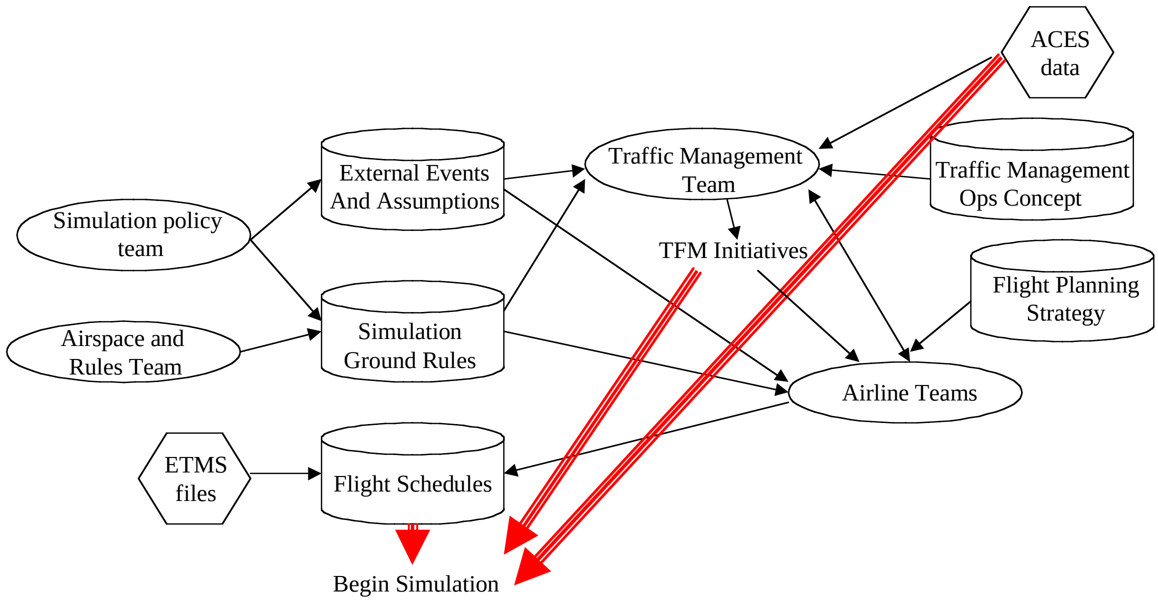


Figure 3-15. The simulation is begun

7. Figure 3-16 shows that as the simulation runs, characteristics are generated to provide status information to the Airline Teams and to the TM Team for their evaluation. This same information is mined to provide simulation results.

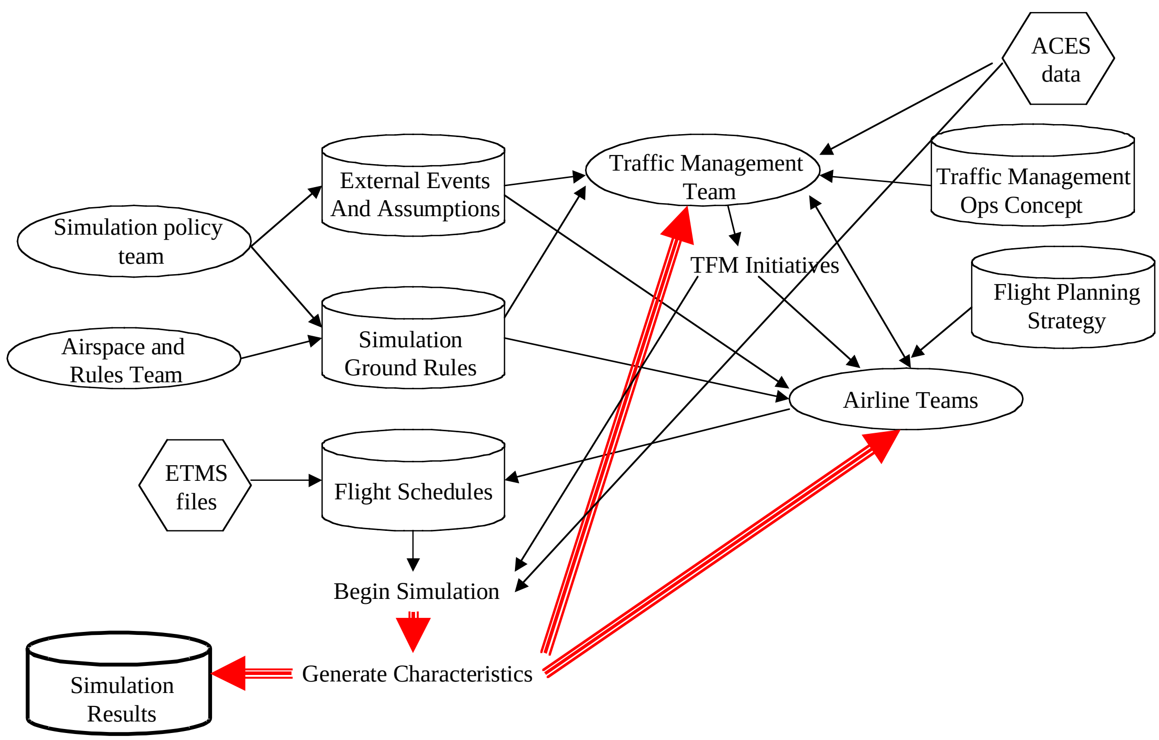


Figure 3-16. System status information is provided to TM and airline teams

8. Figure 3-17 shows that the TM and Airline Teams monitor the simulation state, in light of their individual objectives (derived from their operational concept and flight planning strategy respectively), and negotiate changes.

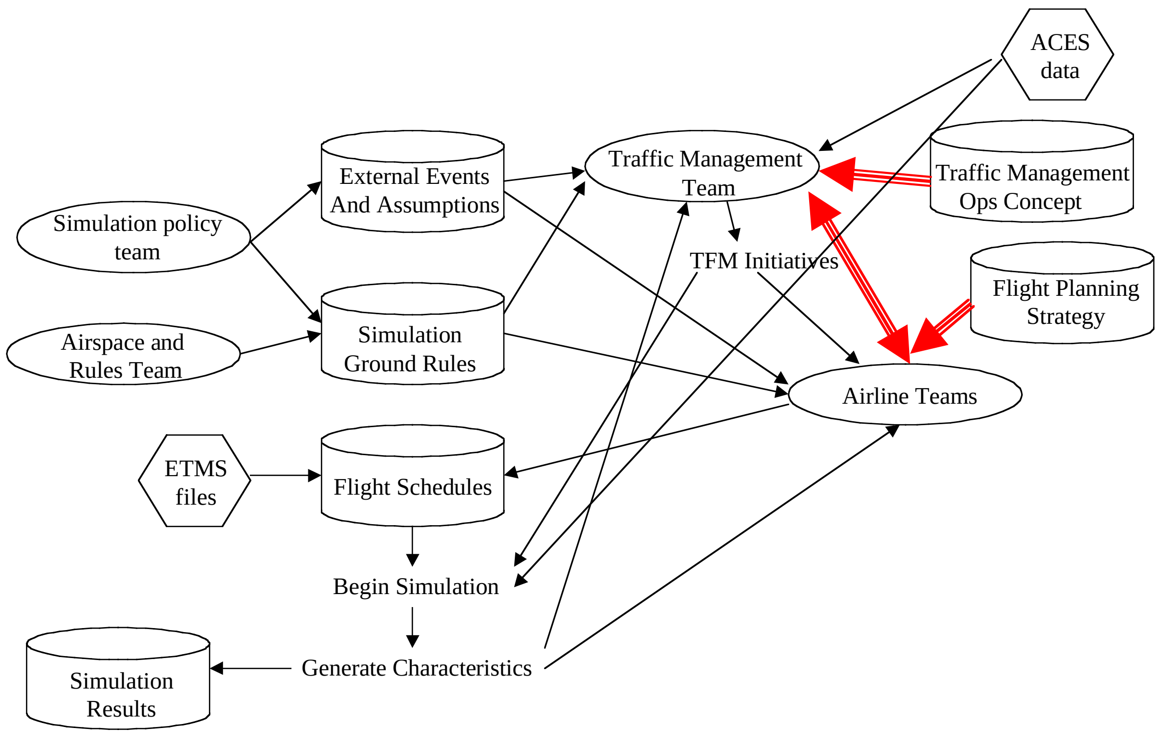


Figure 3-17. TM and Airline Teams monitor simulation state and negotiate changes

9. Figure 3-18 shows that the TM Team makes changes to the TFM Initiatives in place and the Airline Teams make changes to flight plans and to flight schedules. Depending upon the simulation ground rules defined, steps 7 through 9 can be performed on a continuous basis.

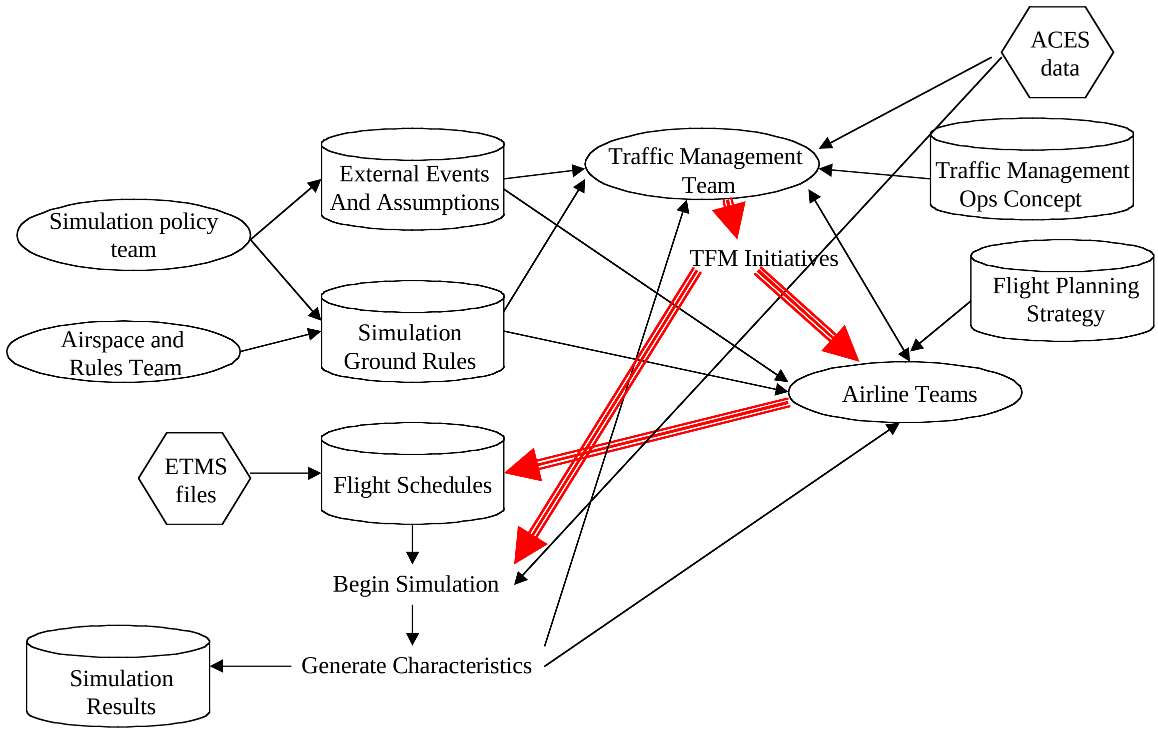


Figure 3-18. The TM and Airline Teams make changes to TFM initiatives and flight schedules

10. In Figure 3-19 the simulation runs to completion and the results are archived for the simulation policy making team and the airspace and rules team to review in full. The Airline Teams and TM Team review their individual results as compared to their objectives.

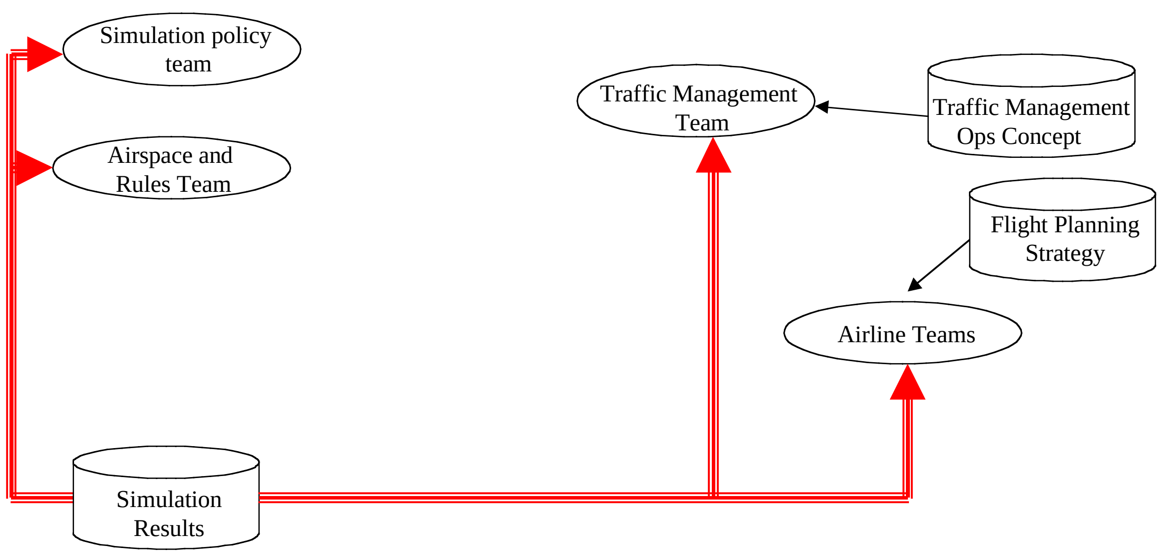


Figure 3-19. The simulation runs to completion and the teams review results

A planned group of scenarios, resulting in a series of simulation runs as described above, will demonstrate the results of strategies defined by all the teams. The review and evaluation of these results may influence the definition of the next set of scenarios. CHATS will allow testing many new concepts and varying many parameters well beyond the capabilities of today's air traffic control system.

Communications and Operational Procedures

Figure 3-20 illustrates the recommended wide-area communications for the distributed simulation. In this concept, a principal internet service provider (ISP) is employed to provide a secure server for the conduct of the simulation. This server connects the Central Operations Complex's LAN with the Internet, and provides secure delivery services through to all the simulation players. The players can use existing ISPs and connection with adequate bandwidth. The System Command Center and the AOCs already have or will soon have adequate Internet connections, which are used for example to view independent weather products.

The principal ISP provides three essential services:

- A secure server
- Installation and management
- 24x7 monitoring

The use of the internet for the CHATS wide area network will minimize the need for NASA to invest in costly communication equipment and line costs.

An alternative to the model of Figure 3-20 would be to use the AOCnet, a private internet-like network that connects the AOCs and the FAA facilities for use in CDM. A connection to this network would have to be made to the Central Operations Complex.

Figure 3-20 focuses on data communications. Voice and messaging communications also would link the players, either separately or over the Internet.

In the operations concept that we have described, the Central Operations Complex keeps master simulation time. The remote players view system status at current time and take action based on this and other communications, or the simulation is run in a "look-ahead" mode to anticipate future conditions. If CHATS were expanded to couple independent models in an automated fashion, for example using an ETMS simulation engine at the System Command Center in parallel with the models described above, the complex issues encountered by the Defense Department in implementing war-gaming exercises would start to become applicable. Appendix B describes how DOD's large-scale simulations handle time management and other issues.

If CHATS is going to be built in an extensible manner to have additional capabilities (see Section 4.3), the DOD's simulation development methodology called High-Level Architecture (HLA) is recommended. Appendix C describes HLA including civilian applications. HLA was evaluated by a joint FAA/Army/NASA team (JFAN) and was considered a useful approach. The advantage of HLA, which can be applied to any communication and model infrastructure, is its ability to handle and resolve potential problems in timing, interface protocols and other areas, problems which arise when bringing independent simulations to work together. Using HLA as a design methodology, even if not fully through the development stage, should minimize future operational problems.

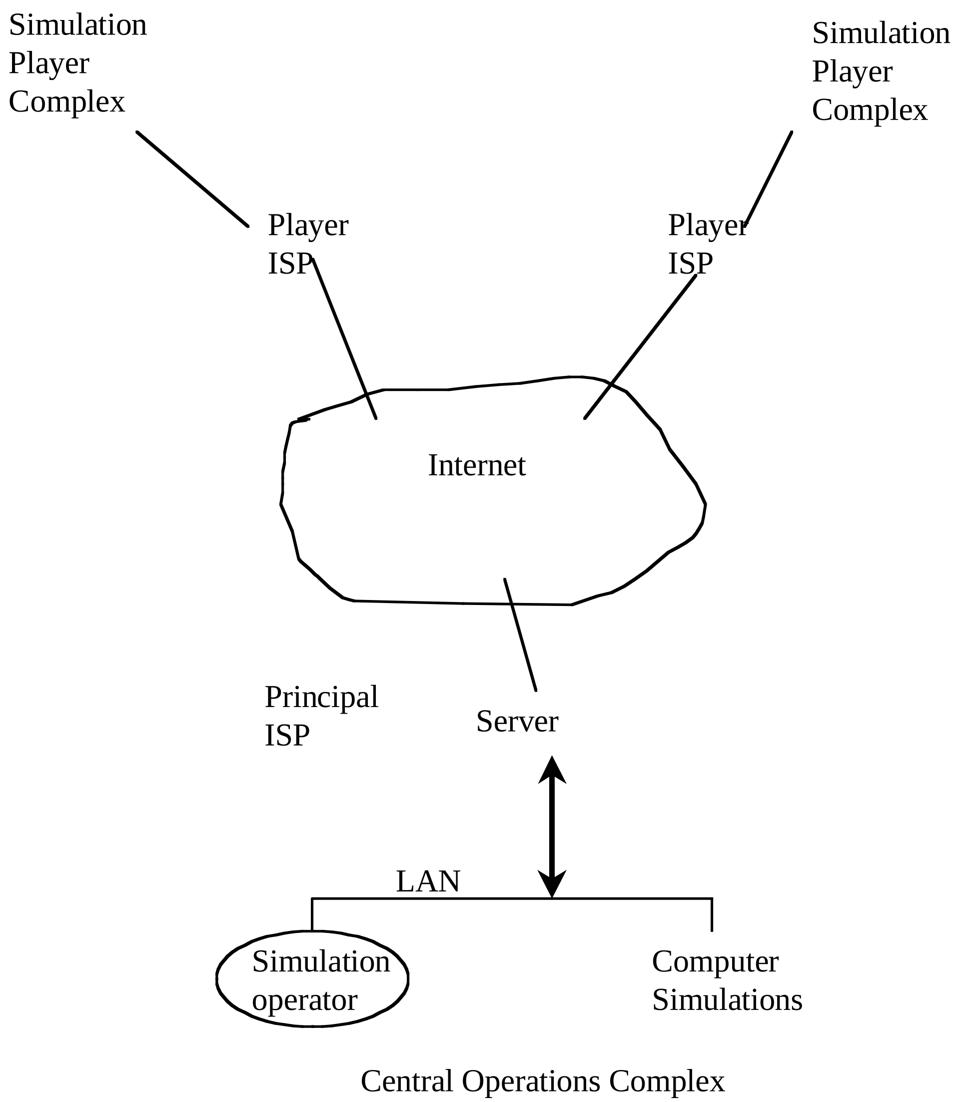


Figure 3-20. Wide-Area Network Schematic

3.3 Problems and Scenarios

The following six problem areas have been defined as examples for CHATS, and scenarios developed for each. The CHATS functional design will handle all of these scenarios.

The scenarios for the six problem areas are oriented toward a future air traffic operational concept which maximizes the discretion and initiative of airlines, and reduces the role of traffic management to monitoring the process and serving as referee. In each case, the same problem could be run in CHATS with today's operational concept for comparison. In fact, this should be done to show the distinctiveness of future concepts and their potential benefits.

Many of these problems and scenarios are similar to activities performed today under collaborative decision making. The reason for running these in CHATS is to test how robust future operational concepts will be in the face of common problems.

1. **Problem:** Initial Scheduling - The initial flight schedules need to be set.

Scenario:

Summary

- Airlines may define new schedules based on their objectives, and designed to an assumption of higher future demand than today. The airlines and traffic management must then negotiate schedules that will work at each airport.

Process Steps

- 1) The airlines send in their proposed flight schedules to traffic management.
- 2) Traffic management does a simulation run "look-ahead".
- 3) Traffic management determines where proposed departures or arrivals at any airport exceeds capacity, and where sector loads exceed capacity, and sends results to the airlines.
- 4) Airlines negotiate changes in their flight schedules with traffic management, and perhaps with each other, until the global schedule set is feasible.

<u>or</u>

- 4) Traffic management assigns slots and airlines are free to assign flights to slots as they choose, and perhaps negotiate buy-sell-trade slots with each other.

2. **Problem:** Departure Sequencing - The pre-determined departure sequence has to be changed.

Scenario:

Summary

- A departing flight F of airline A is delayed and cannot make its pre-assigned takeoff slot. The airlines re-negotiate a departure sequence.

Process Steps

- 1) Airline A proposes to swap slots between F and another of its own flights G. In all cases the airline notifies traffic management and must obtain permission.

- 2) As an alternative, airline A re-arranges a bank of flights to fit existing slots and all constraints and proposes the result.
- 3) As an alternative, airline A negotiates a slot swap with airline B for flight F and proposes the result

3. **Problem:** Approach and Landing Sequencing - The pre-determined approach and landing sequence has to be changed.

Scenario:

Summary

- An arriving flight F of airline A is delayed and cannot make its pre-assigned approach slot. The airlines re-negotiate an approach sequence.

Process Steps

- 1) Airline A proposes to swap arrival sequence between F and another of its own flights G. In all cases the airline notifies traffic management and must obtain permission.
- 2) As an alternative, airline A re-arranges more than one flight to fit existing sequence positions and all constraints and proposes the result.
- 3) As an alternative, airline A negotiates a sequence swap with airline B for flight F and proposes the result.

4. **Problem:** Intermediate Weather Diversion - A weather problem affects all flights passing through a defined geographic area over a certain time period. There may be uncertainty as to when the front will come through.

Scenario:

Summary

- Airlines re-plan their flights that were scheduled to pass through the bad weather area.

Process Steps

- 1) Airlines prepare and submit new trial flight plans for launched flights as necessary to avoid bad weather area.
- 2) Traffic management does a simulation run “look-ahead” and may deny a group of flight plans due to potentially overloaded sectors.
- 3) Airlines submit alternate flight plans until all are approved.
- 4) Airlines do ground holds and cancellations as necessary for future flights.

5. **Problem:** Restricted Arrival Rate - ATC cuts the allowed arrival rate due to bad weather or technical problems.

Scenario:

Summary

- ATC cuts the allowed arrival rate for each airline into airport k by 50%, due to the need to close one of two parallel runways.

Process Steps

- 1) ATC takes away half the slots from each airline over a defined period of time, or into the indefinite future, and applies ration-by-schedule.
- 2) Each airline selects 1/2 of its planned arrivals to airport k to enter available slots, including if possible all airborne flights at the time of the change. Attempts to meet required time of arrival (RTA) constraints as much as possible.
- 3) The remaining flights are cancelled or delayed in the hope that the restriction will be lifted.

This scenario can be expanded to cover several airports simultaneously affected by bad weather.

6. **Problem:** Aircraft Out of Service - An airplane intended for service on an upcoming flight segment is diverted for maintenance.

Scenario:

Summary

- Airline A must decide whether or how to serve flight F, originally scheduled to be served by an aircraft which has been taken out of service.

Process Steps

- Airline A examines alternatives:
  - cancel flight F
  - re-assign another aircraft to flight F thereby requiring cancellation or delay of flight G
  - ferry spare aircraft to fly delayed flight F
- Re-arranges schedule according to decision.
- Proceed as in departure sequencing problem area.

3.4 Simulation Result Indicators and Metrics

The following indicators and metrics for evaluating the results of the simulations are suggested. The indicators address *outcomes* to the maximum extent possible, that is results from the perspective of each stakeholder, and the metrics measure changes in the indicators. If an indicator is too difficult to measure directly, indirect metrics are defined.

The indicators are divided into the following three areas:

- Quantitative indicators related to system operational performance, and of interest to the FAA in NAS oversight;
- Quantitative indicators related to air carrier performance; and
- Qualitative indicators capturing subjective reports by the players concerning strategies used and opinions of different operational concepts.

The proposed indicators and metrics should be considered a starting point, and it is expected that additional ones will be defined during CHATS development. In particular, the traffic management team and the user teams are expected to define indicators and metrics that relate to their objectives.

System Operational Performance Indicators

The FAA's Air Traffic Service organization (FAA, 1997(1)) has defined seven indicators of system operational performance. For six of these indicators, metrics are defined as they can be produced as evaluation results from CHATS.

<u>Safety</u>
Conflicts requiring controller action can be measured in CHATS.
- Number of conflicts requiring controller action as a % of operations
- Frequency distribution of conflicts by sector and time of day

<u>Delays</u>
Delays can be measured in relation to <u>scheduled</u> time, and also to <u>optimal</u> time from the OPGEN model.
- % distribution of flights by minutes of delay

<u>Flexibility</u>
This indicator is addressed by comparing outcomes of different sets of rules which differ in flexibility allowed to the airlines, e.g.
- fixed routes vs. user-preferred routes
- 200 nmi limit for user-preferred routes, vs. no limit

<u>Predictability</u>
- Variance of actual from scheduled flight times
- % flights cancelled or diverted

<u>Access</u>
Access is defined as a user's ability to use air traffic services. It is not addressed in CHATS.

<u>Availability</u>
This indicator is addressed by comparing outcomes of different scenarios of NAS system equipment or information unavailability.

<u>Productivity</u>
This indicator concerns service provider productivity and workload.
- % distribution of sector loads. The sector load is measured as the % of maximum allowable aircraft in the sector from a controller standpoint.

Air Carrier-Specific Indicators

Six areas have been proposed for performance from an air carrier perspective (CNS/ATM Focused Team, 1998). Metrics have been defined for five of these indicators.

Delays
The same measure as above, but for the individual air carrier.

Flexibility
The same measure as above, but for the individual air carrier.

Predictability
For the individual air carrier:
- Variance of actual from scheduled flight times
- % flights cancelled or diverted
- % of flights in arrival banks which cause missed connections with a flight in a departure bank

Access
This indicator is not addressed in CHATS.

Efficiency
Efficiency addresses means of reducing an air carrier's direct operating costs through, for example:
- Optimized flight path trajectories
  - Distribution of scheduled/optimum flight times
  - Distribution of actual/optimum flight times
- More or less banking of flights

Cost
For the individual air carrier:
- Average fleet utilization
- Fuel usage (normalized)

Qualitative Indicators

A method for eliciting subjective feedback from all the players will be created. The following questions are expected to be included:
- Were there problems in carrying out a scenario? Which of these problems would be likely to occur also in the real world?
- Was the operational concept successful? What problems occurred due to the concept?
- What strategy did your team employ in this scenario?
- What was successful, and what was not, in your carrying out of the strategy?
- Did stakeholder feedback, for example poor achievement of team objective, lead to policy changes for following simulation runs?

4.0 CHATS Development Plan

4.1 Development Strategy

A series of steps has been laid out for the development and operation of CHATS. These are used as a basis for planning, scheduling, and costing the job.

The development steps A through E below are designed to first develop a rapid prototype, then start simulations with the prototype, then expand the prototype into a fully operating system, then run additional scenarios and evaluate the results. These are illustrated with a schedule in Figure 4-1.

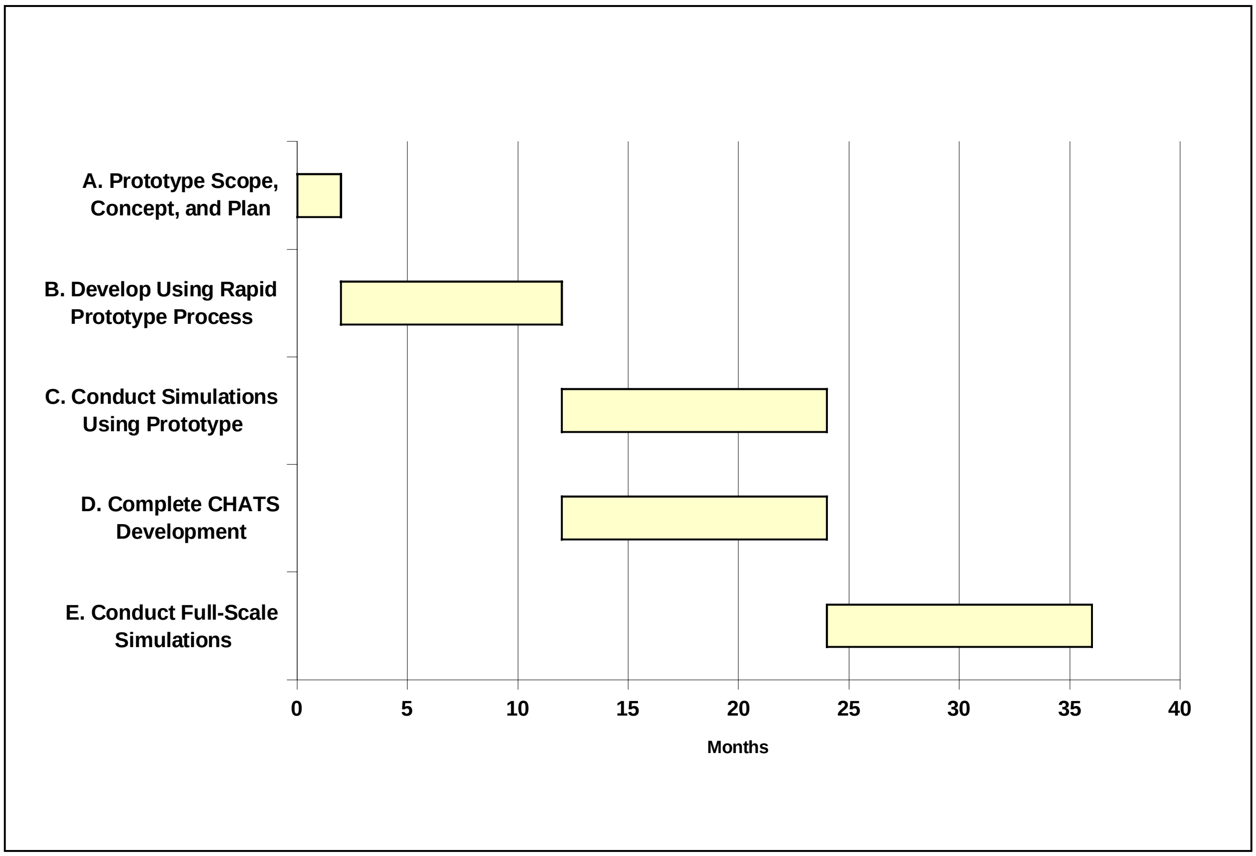


Figure 4-1. Timeline for CHATS Development and Operation

A. Prototype Scope, Concept, and Plan

During this step, we establish requirements, scope and an operational concept for a reduced functionality CHATS prototype, and plan the prototype development. The planned scope of the prototype is to simulate the interaction of a single AOC and the FAA's ATCSCC, with colocated workstations. Useful tests and scenarios will be run using the prototype while the full system is being developed.

1) Establish requirements, scope and operational concept for the prototype

2) Plan a series of incremental, increasing functionality for the prototype

Resources required:

Planning and system engineering personnel.

B. Develop Using Rapid Prototype Process

Develop the prototype system.

1) Perform a continuing series of cycles of build, test, evaluate, modify, and add capability until the prototype functionality reaches the desired level. Each cycle would have the following steps:

1.1) Acquire necessary hardware and facilities for the planned function

1.2) Modify existing software packages and develop new software as necessary

1.3) Perform system integration including hardware and software modules

1.4) Prepare test plan, conduct functional and operational tests of prototype system and evaluate performance

1.5) Modify system as necessary based on evaluation

1.6) Re-determine future planned increments for the prototype

2) Prepare operations and user guides to the conduct of simulations using the prototype

Resources required:

- Planning and directing personnel
- System engineering personnel
- Hardware, platforms, networks, displays
- Physical space
- Software developers
- Test and operations personnel

- Stakeholder evaluation personnel
  - airline
  - traffic management
  - other
- Documentation personnel

<u>C. Conduct Simulations Using Prototype</u>

Run simulations over a period of time, using scenarios that focus on information flow between an AOC and the ATCSCC. Do an evaluation report on the results.

All teams can be involved in scenarios using the prototype, including the concept definition and experiment design team.

1) Prepare master plan for conduct of simulations

2) Perform the following sequence a number of times according to the plan:

2.1) Plan a simulation run or a sequence of simulation runs

2.2) Run all simulations within current plan

2.3) Evaluate results of these simulations

After all simulation runs have been completed:

2.4) Prepare overall evaluation report on the results of the simulations

Resources required:

- Planning and directing personnel
- Test and operations personnel
- Stakeholder personnel participating as players
  - airline
  - traffic management
  - other
- Documentation personnel

<u>D. Complete CHATS Development</u>

Expand the prototype to achieve the full planned system. The expansion will be to the scope and functional requirements described in this report, namely a distributed system with several user teams and the ATCSCC linked to a central operations complex.

1) Document the as-built design of the prototype system

2) Validate the requirements for full desired CHATS capability laid out at the start of development and change as necessary

3) Modify the prototype design to represent full system capability

4) Acquire incremental hardware and facilities to add to the prototype

5) Modify prototype software base and develop new software as necessary

6) Perform system integration

7) Prepare test plan, conduct functional and operational tests and evaluate performance

8) Modify system as necessary based on evaluation to meet functional and operational requirements

9) Update operations and user guides to the conduct of simulations

Resources required:

- Planning and directing personnel
- System engineering personnel
- Hardware, platforms, networks, displays
- Physical space
- Software developers
- Test and operations personnel
- Stakeholder evaluation personnel
  - user teams
  - traffic management
  - other
- Documentation personnel

<u>E. Conduct Full-Scale Simulations</u>

Run simulations over a period of time, using the full range of planned scenarios. Do an evaluation report on the results.

The steps are the same as described under C. above, Conduct Simulations Using Prototype.

Resources required:

- Planning and directing personnel
- Test and operations personnel
- Stakeholder personnel participating as players
  - user teams
  - traffic management

    - other
- Documentation personnel

Timeline for Development

Figure 4-1 is a timeline showing the following steps:

- Development of a rapid prototype in the first year. This consists of steps A, Prototype Scope, Concept and Plan (2 months), and B, Develop Using Rapid Prototype Process (10 months).
- Running useful scenarios on the prototype, concurrent with expansion to the full system, in the second year. This consists of steps C, Conduct Simulations Using Prototype (12 months), and concurrently D, Complete CHATS Development (12 months).
- Running scenarios with the full system in the third year and providing an overall evaluation of results.

4.2 Cost Estimate

Table 4-1 presents a cost estimate for CHATS. The detail behind these costs is shown in Appendix D, along with the assumptions that were made in deriving them.

Table 4-1 CHATS Cost Estimate

	Development		Operation		
	Labor-Years	Cost ($K)	L-Y	Cost ($K)	Total
A. Prototype Scope, Concept and Plan	0.3	50			
B. Develop Using Rapid Prototype Process	3.0	500			
C. Conduct Simulations Using Prototype			2.1	340	
D. Complete CHATS Development	3.1	590			
E. Conduct Full-Scale Simulations			3.7	690	
Total	6.4	1140	5.8	1030	2170

The total life-cycle cost for the three-year program is slightly over $2 million, evenly divided between development and operation.

Items A and B together would make up a first-year program costing $550K, producing a ready-to-run prototype system.

4.3 Extensions of CHATS Functionality

The following stages of development are proposed as optional future increased capabilities for CHATS. The strategy behind this development sequence is to first increase fidelity with respect to airline strategic decisions, then expand to the tactical arena which is already being simulated on other NASA platforms.

Within each stage the following three steps would be conducted:

<u>A. Develop New Requirements and Modify Design</u>

<u>B. Complete Development</u>

<u>C. Run Simulations</u>

Add Air Carrier Economic Models

Provide economic models to represent long-term business planning as decision support within the simulation. With these models, carrier strategies such as serving new airports, establishing/changing hub locations, and new aircraft types would have increased credibility.

Add Representation of Terminal Areas and Airports

Add increased fidelity in selected terminal areas and airports to represent decisions made there in more detail.

Add Conflict Alert /Resolution Procedures

Add representation of conflict alert/avoidance/resolution procedures with pilots and controllers as added simulation players. Accomplish this by developing an interface with other NASA simulations relating to distributed air/ground decisionmaking.

5.0 Summary, Conclusions and Recommendations

An operational concept and a functional design have been created for CHATS. The system, and the simulations to be run using it, will focus on aviation user strategic decision-making in a free flight environment.

CHATS will employ the services of the following decision makers:

- Air Carriers – to make strategic decisions concerning the management of their fleets, in particular to meet their schedules in the face of obstacles, both prior to and during the conduct of a simulation run.
- Military and general aviation users – to observe impacts on their class of flights, devise strategies and provide feedback to the policy team.
- Traffic management – to make strategic decisions concerning the management of the National Airspace System, in particular resolving bottlenecks and overloads, both prior to and during the conduct of a simulation run.
- Airspace and rules specialists – to create new concepts of organizing the airspace and air traffic rules, consistent with potential future free flight visions, to be utilized in the simulations.
- A simulation policy body – to manage the CHATS resource by creating sets of scenarios which will explore significant future free flight operational concepts.

User and traffic management teams will also serve as human players during the conduct of the simulation runs. The simulation will be under the control of an operator who will create a detailed script for each defined scenario and implement it in a simulation run.

The computer simulation within CHATS will be based on existing air traffic models. NARIM will be used to generate the 4-D aircraft trajectories within the continental United States. ETMS will be used as a data base and presentation tool for part of the system status during the conduct of the simulation. FSM can also be used in the presentation of airport-related information. A queuing model to take account of airport capacity constraints must be developed, or an existing airport queuing model adapted for use. Extensions to NARIM functionality need to be developed along with new system status elements for the players. The new required system and software should be easy to develop given the basic functions which are already in place.

CHATS is designed as a distributed system, with a central operations complex and user teams, traffic management, and potentially other decisionmakers using remote workstations at their normal places of business. This will minimize travel time for these personnel. The attention they give to the simulation could be on an as-needed basis. A simulation run would generally be conducted in fast time with pauses to examine the situation and decision points. However, the system will be capable of running a simulation in real time for some defined period, for greater realism.

CHATS will be able to run scenarios which introduce external problems, such as bad weather, airport capacity reductions, SUA activation, and out-of-service aircraft. It will handle initial flight schedules, flight plans, and modifications to flight plans. Air carriers will be able to modify their fleet deployment based on the course of events in the simulation. Negotiation between air carrier and traffic management players, and among air carriers, will be a major activity during a simulation run.

More fundamentally, air carriers will be able to propose wholly revised schedules and show their implications. Also the simulation policy team can introduce new airports or airport expansions.

A development plan has been presented which will build and run CHATS for a little over $2 million, over a period of three years. During the first year a rapid prototype will be built with an operations center, a traffic management workstation, and an AOC workstation all colocated on a local area network. During the second year, the prototype will be utilized in an operational mode to examine traffic management – AOC interactions and communications, while the full system is being constructed including the distributed locations. During the third year, the full system will be utilized in an operational mode. The $2 million life cycle cost is evenly divided between development and operations.

It is recommended that NASA go ahead with CHATS, which will explore user competitive and cooperative behavior in a free-flight environment, areas not being addressed elsewhere within the aviation community. A first-year program which would create a ready-to-run prototype system would cost $550K.

6.0 References

CNS/ATM Focused Team (1998), Report of the Air Traffic Services Performance Focus Group, October.

FAA (1997(1)), Air Traffic Services Performance Plan for Fiscal Years 1997-1999.

FAA (1997(2)), ATS Concept of Operations for the National Airspace System in 2005: Narrative, September.

RTCA Inc. (1995), Final Report of RTCA Task Force 3, Free Flight Implementation, October.

SRC (1998), Feasibility Study and Requirements Definition for CHATS, Phase I Report, July.

Appendix A. Phase I Desired Capabilities

Table A-1, CHATS Desired Capabilities and Requirements, shows the capabilities determined from Phase I as modified for the current concept. Eliminations are shown in strikeout format, and text additions in underline format. The following describes each change in order of the numbering in the table.

1.1.4 With CHATS' strategic focus, human-in-the-loop capability for pilots and controllers is not needed. Alternative NASA programs are providing that capability.

1.1.7 Live traffic data from a facility is more detail than needed in the model; ETMS inputs will be sufficient, since CHATS has a simplified representation of terminal areas and airports.

2.2.1 Same comment as for 1.1.4.

3.1 and 3.1.1 Since the NARIM model can simulate CONUS, we will do so. The simulation is not detailed in the sense that terminal areas and airports have a simplified representation.

6.2.1 With the focus of this report, strategy teams can be limited as shown.

8, 9, 10, 11. These capabilities require air carrier economic models. To create a credible match between the simulation and an air carrier model would be too big an effort for the concept described in this report, and is better suited to a later expansion.

19, 20, 21. With the user strategic focus, conflict detection and resolution are not addressed in detail. Alternative NASA programs are providing that capability, which could later be interfaced with CHATS.

24. Same comment as for 8.

25. Technology requirements are driven by safety and information needs to implement new traffic management concepts. CHATS is examining air traffic under the assumption that safety is maintained and that timely information is available to air carriers, pilots and controllers. It is not a good guide to technology needs.

Table A-1. CHATS Desired Capabilities and Requirements

Stated CHATS Capabilities	**Derived Operational Requirements (The CHATS system and/or its users shall...)**	**Derived Functional Requirements (The CHATS system and/or its users shall...)**
1. Establish a capability to simulate long-term system operation and adaptation	1.1 Provide a digital simulation capability for the major elements of the NAS and its functions from the existing system to beyond the NAS Architecture Version 3.0 (i.e., 2015 and beyond)	1.1.1 Provide a digital simulation capability of NAS airspace (all classes). 1.1.2 Provide a digital simulation of the communications, navigation, surveillance, automation, and weather functions of the NAS 1.1.3 Provide a digital simulation of the existing, mature state and alternative traffic management concepts of operation 1.1.4 Provide human-in-the-loop capability for dispatchers, and system stakeholders. Provide capability for directed route changes based on weather, congestion, delay & other factors. 1.1.5 Be able to simulate weather and other disturbances to the normal modes of operation 1.1.6 Have a pause and resume capability as well as real-time and faster-than-real-time capability 1.1.7 Allow traffic scenarios including a mix of different aircraft, avionics equipage, and user classes to drive the simulation. Capability to accommodate inputs from the ETMS shall also be possible 1.1.8 Provide simulations of all aircraft, flight crews, flight crew awareness, pilots, controllers, tactical decision aids, and conflict detection and resolution. 1.1.9 Provide data recording and playback capability. Incorporate on-line, quick look and post simulation data reduction and analysis capabilities

Stated CHATS Capabilities	Derived Operational Requirements (The CHATS system and/or its users shall...)	Derived Functional Requirements (The CHATS system and/or its users shall...)
2. Combine computer-assisted cooperative work methodology and new or existing simulations	2.1 Develop CHATS making maximum use of existing facilities	2.1.1 Provide a centralized CHATS capability with a network capable of connecting with two way data and voice communications to applicable existing simulators/simulations
	2.2 Provide users with local workstations linked to a network that ties the CHATS elements together.	2.2.1 Provide local user interactive capability with the CHATS. Communications capabilities among and between dispatchers, and stakeholders shall be provided when man-in-the-loop simulation is used and simulated when not used.
3. Determine the improvements in capacity, safety, and user flexibility of a distributed responsibility traffic management system vs. a centralized planning and control system	3.1 Simulate a challenging airspace (e.g., CONUS) using the existing centralized planning and control system CONOPS to evaluate present capacity, safety, and flexibility metrics	3.1.1 Provide a simulation of a challenging airspace (e.g., CONUS) including all existing routes, waypoints, approaches, departures, airports, ATCTs, TRACONs, ARTCCs, AOCs and existing letters of agreement with adjacent facilities.
		3.1.2 Provide a realistic mix of traffic, weather and other disturbances for the selected area.
		3.1.3 Provide the capability to evaluate metrics related to capacity, delay, safety, efficiency, throughput, and flexibility (Collect and process simulation data)
	3.2 Simulate the same airspace as in 3.1 using a distributed responsibility traffic management system to evaluate capacity, safety, and flexibility metrics	3.2.1 Provide the capabilities described in 3.1.1, 3.1.2 and 3.1.3 for a distributed responsibility traffic management system operating in the same airspace
	3.3 Compare distributed vs. centralized traffic management concepts to determine the capacity, safety, and flexibility differences	3.3.1 Provide the capability to process and compare the results of the simulation for several runs of different traffic management concepts
4. Determine the sensitivity of free-flight concept benefits to varying levels of necessary user equipage and of system equipment failures	4.1 Determine free-flight concept benefits	4.1.1 Simulate the free-flight concept in the same manner as requirement 3.2.1 and then compare the results as in 3.3.1 with other traffic management concepts to determine the improvements in the metrics (i.e., benefits) related to capacity, delay, safety,

Stated CHATS Capabilities	Derived Operational Requirements (The CHATS system and/or its users shall...)	Derived Functional Requirements (The CHATS system and/or its users shall...)
		efficiency, throughput, and flexibility
	4.2 Determine sensitivity of free-flight concept benefits to airborne user equipage adoption	4.2.1 Simulate different levels of airborne user equipage adoption and reevaluate as in 4.1.1
	4.3 Determine sensitivity of free-flight concept benefits to airborne equipment failures	4.3.1 Simulate all combinations of possible airborne equipment/software failures and reevaluate as in 4.1.1
	4.4 Determine sensitivity of free-flight concept benefits to ground based equipment failures	4.4.1 Simulate all combinations of possible ground base (equipment, software) failures and reevaluate as in 4.1.1
5. Determine the sensitivity of free-flight concept benefits to operator noncompliance with voluntary system rules	5.1 Determine voluntary system rules for free-flight	5.1.1 Synthesize a preliminary set of voluntary rules for free-flight based on stakeholder inputs
		5.1.2 Simulate the free-flight system to evaluate the utility of the preliminary set of voluntary rules for free-flight and modify and validate as needed via simulation and evaluation of system metrics. Differing rules may apply depending on the airspace being considered.
	5.2 Measure how free-flight concept benefits vary with degrees of operator noncompliance with voluntary system rules specified in 5.1	5.2.1 Simulate the free-flight system using the set of voluntary rules for free-flight derived in 5.1.2. Run various scenarios with operator noncompliance to one or more of the rules and evaluate the impact on the system metrics
6. Estimate system stakeholder tendencies to compete and cooperate	6.1 Identify stakeholders	6.1.1 Identify stakeholders based on all entities with an interest in the NAS services including pilots, controllers, GA, airlines, military, cargo, air taxi, helicopters and the like.
	6.2 Configure stakeholder strategy teams	6.2.1 Create strategy teams with members (either active or retired) from the following stakeholder groups: airspace users, traffic management, and FAA airspace and rules. A cadre of members for each group will be identified in order to assure availability when needed. Certain simulations will require only specific groups

Stated CHATS Capabilities	Derived Operational Requirements (The CHATS system and/or its users shall...)	Derived Functional Requirements (The CHATS system and/or its users shall...)
		while others may require all groups.
	6.3 Integrate stakeholders into the free-flight simulation	6.3.1 Develop models of stakeholder strategy group objectives for incorporation into the CHATS in order to minimize the need for live interaction with the simulation. Some simulation experiments will however require live stakeholder strategy teams to interact directly in real time with the simulation. 6.3.2 Develop a model for negotiations between stakeholder strategy teams for incorporation into the CHATS 6.3.3 Incorporate methods for strategy teams to design, implement, and modify strategic decision making logic and traffic management procedures and airspace restrictions. An option for keeping the logic for one strategy team secret from other strategy teams will be available.
	6.4 Measure degrees of competition and cooperation for defined stakeholders participating in the free flight simulation	6.4.1 Identify metrics to evaluate the cooperation and competition among stakeholders 6.4.2 Conduct simulations to evaluate the competition and cooperation metrics under a variety of traffic and airspace conditions.
7. Determine if stakeholder competitive actions could jeopardize concept feasibility	7.1. Determine impact of stakeholder competitive actions on capacity, safety, and flexibility	7.1.1 Conduct simulations to evaluate impact of cooperation and competition on the capacity, safety, and flexibility metrics under a variety of traffic and airspace conditions

Stated CHATS Capabilities	Derived Operational Requirements (The CHATS system and/or its users shall...)	Derived Functional Requirements (The CHATS system and/or its users shall...)
12. Determine the equitability of investigated concepts for each user class	12.1 Define the metric “equitability” 12.2 Estimate “equitability” for GA, Air Carrier, and Military user classes and categories within each class for various alternative distributed, centralized and collaborative traffic management concepts	 12.2.1 Calculate the metrics for “equitability” via simulation of GA, Air Carrier, and Military user classes and categories within each class for various alternative distributed, centralized and collaborative traffic management concepts
13. Determine which regions, conditions, airspace classes and traffic levels are appropriate for free-flight/ free maneuvering	13.1 Identify a set of discrete regions, conditions, airspace classes and traffic levels that are representative of the spectrum of conditions encountered in the NAS by the various user classes 13.2 Evaluate free-flight performance in each of the environments identified in 13.1 13.3 Determine in which regions free flight performance is “acceptable”	13.1.1 Develop simulation scenarios and cases for the se of conditions identified in 13.1 13.2.1 Conduct simulations of free flight using the scenarios developed in 13.1.1 and evaluate the impact o the set of system metrics previously established. 13.3.1 Determine acceptable levels of performance and evaluate the results of the free-flight simulations against these levels to identify those regions, conditions, airspace classes and traffic levels that are appropriate fo free-flight maneuvering
14. Determine which regions, conditions, airspace	14.1 Evaluate positive ground-based control in each of	14.1.1 Conduct simulations of positive ground-based

Stated CHATS Capabilities	Derived Operational Requirements (The CHATS system and/or its users shall...)	Derived Functional Requirements (The CHATS system and/or its users shall...)
classes and traffic levels are appropriate for positive ground-based control	the environments identified in 13.1	control using the scenarios developed in 13.1.2 and evaluate the impact on the set of system metrics previously established.
	14.2. Determine in which regions positive ground-based control performance is “acceptable”	14.2.1 Determine acceptable levels of performance and evaluate the results of the positive ground-based control simulations against these levels to identify those regions conditions, airspace classes and traffic levels that are appropriate for positive ground-based control
15. Determine which regions, conditions, airspace classes and traffic levels are appropriate for dynamic route structures	15.1. Evaluate the use of dynamic route structures in each of the environments identified in 13.1	15.1.1 Conduct simulations of dynamic route structures using the scenarios developed in 13.1.2 and evaluate the impact on the set of system metrics previously established.
	15.2 Determine in which regions the use of dynamic route structures is “acceptable”	15.2.1 Determine acceptable levels of performance and evaluate the results of simulations of the use of dynamic route structures against these levels to identify those regions, conditions, airspace classes and traffic levels that are appropriate for the use of dynamic route structures
16. Determine which regions, conditions, airspace classes and traffic levels are appropriate for dynamic sectorization	16.1 Evaluate the use of dynamic sectorization in each of the environments identified in 13.1	16.1.1 Conduct simulations of dynamic sectorization using the scenarios developed in 13.1.2 and evaluate the impact on the set of system metrics previously established.
	16.2 Determine in which regions the use of dynamic sectorization is “acceptable”	16.2.1 Determine acceptable levels of performance and evaluate the results of simulations of the use of dynamic sectorization against these levels to identify those regions, conditions, airspace classes and traffic levels that are appropriate for the use of dynamic sectorization
17. Determine which regions, conditions, airspace	17.1 Identify other NAS modernization concepts	17.1.1 Collect and analyze AATT concepts, FAA

Stated CHATS Capabilities	**Derived Operational Requirements (The CHATS system and/or its users shall...)**	**Derived Functional Requirements (The CHATS system and/or its users shall...)**
classes and traffic levels are appropriate for other NAS modernization concepts		concepts of operations, academic, industry and other views on NAS modernization
	17.2 Evaluate the use of other NAS modernization concepts in each of the environments identified in 13.1	17.2.1 Conduct simulations of other NAS modernization concepts using the scenarios developed in 13.1.2 and evaluate the impact on the set of system metrics previously established.
	17.3 Determine in which regions the use of other NAS modernization concepts is “acceptable”	17.3.1 Determine acceptable levels of performance and evaluate the results of simulations of the use of other NAS modernization concepts against these levels to identify those regions, conditions, airspace classes and traffic levels that are appropriate for the use of other NAS modernization concepts
18. Evaluate mutually acceptable strategies for cooperation and competition among system stakeholders	18.1 Identify stakeholder cooperation strategies	18.1.1 Define a set of simulation scenarios that will focus on the identification of the various stakeholder teams’ cooperation strategies in differing airspace and traffic situations. Conduct the indicated simulations
	18.2 Identify stakeholder competitive strategies	18.2.1 Define a set of simulation scenarios that will focus on the identification of the various stakeholder teams’ competitive strategies in differing airspace and traffic situations. Conduct the indicated simulations
	18.3 Determine mutually acceptable strategies among system stakeholders	18.3.1 Identify similar strategies among stakeholders based on the results of 18.1 and 18.2.
22. Determine which airspace restrictions are no longer necessary	22.1 Identify the restrictions that exist in the current NAS	
	22.2 Identify the restrictions needed in future centralized traffic management systems	22.2.1 Use expert opinion to propose which restrictions may be needed in future centralized traffic management

Stated CHATS Capabilities	Derived Operational Requirements (The CHATS system and/or its users shall...)	Derived Functional Requirements (The CHATS system and/or its users shall...)
		systems and then verify this judgment by simulation
	22.3 Identify the restrictions needed in future distributed traffic management systems	22.3.1 Use expert opinion to propose which restrictions may be needed in future distributed traffic management systems and then verify this judgment by simulation
	22.4 Evaluate which restrictions can be eliminated by the various traffic management concepts	22.4.1 Compare the results of 22.1, 22.2 and 22.3.
23. Determine optimal separation assurance criteria	23.1 Identify the metrics that define the concept of "optimal" (e.g., safety, capacity, delay, flexibility) separation assurance criteria	23.1.1 Define and develop the metrics for "optimal" separation assurance criteria. Define simulation scenarios that will allow the evaluation of these metrics.
	23.2 Evaluate current system separation assurance criteria	23.2.1 Conduct simulations based on the scenarios of 23.1.1 to evaluate the separation assurance criteria metrics for the current system. Fast-time simulation may be possible for this research.
	23.3 Evaluate separation assurance criteria for alternative future system traffic management concepts	23.3.1 Conduct simulations based on the scenarios of 23.1.1 to evaluate the separation assurance criteria metrics for alternative future system traffic management concepts. Fast-time simulation may be possible for this research.
	23.4 Select the "best" separation assurance criteria for each future concept	23.4.1 Perform data processing and analysis based on the results of the simulations conducted in 23.2.1 and 23.3.1 This will require a very comprehensive data recording/collection, reduction and analysis capability.

Stated CHATS Capabilities	Derived Operational Requirements (The CHATS system and/or its users shall...)	Derived Functional Requirements (The CHATS system and/or its users shall...)

Appendix B. Department Of Defense (DoD) Experience In Large-Scale Modeling And Simulation

B.1 Background And Objectives

The DoD has invested significant resources in the development and exploitation of modeling and simulation (M&S) technologies. Within the DoD acquisition process, simulations have in some cases proven to be an effective tool for lowering weapon systems procurement costs. At best simulations are relied upon to evaluate emerging technologies, analyze future tactics, assist with training, and explore concepts of operation. At worst they offer poor interoperability, do not portray the full range of military operations, and require extensive personnel support. Because each of the DoD Services (Air Force, Army, Navy, Marines) has historically developed M&S systems independently, their emphasis was on key capabilities mirroring Service warfare domains. This made sense when the intended purpose of the simulation was, for example, Service training. The aim of each Service was to create a simulated battlespace that reflected their viewpoint. These independently or "stovepipe" developed systems were adequate at times for Single service training, but a problem occurs when the Services' ultimate desire is to train as they fight. Wars are not fought by individual services, but are fought on the basis of unified action. Therefore a representation of joint military activities was desired and the Joint Simulation System (JSIMS) program was created in an attempt to link existing service simulations for a more accurate portrayal of wargaming scenarios. In creating this joint exerciser certain obstacles must be addressed. What was once a standalone simulation package must now execute in a distributed environment, information and events particular to one service are required by some or all the services, data sources must be shared, and time characteristics must somehow be preserved. This paper attempts to address each of these obstacles and frequently references the JSIMS approach to overcoming these challenges.

B.2 AWSIM And NASM Relationship To JSIMS

As previously mentioned, each of the services has a simulation package with adequate modeling to train within the service. Within the Air Force the Air Warfare Simulation (AWSIM) was the primary simulation package for wargaming and training. To address AWSIM deficiencies as well as to have a simulation package operate alone or as a member of the JSIMS family for joint wargaming the Air Force has the requirement to create the National Air and Space (Warfare) Model (NASM). NASM extends and replaces the AWSIM to provide an operationally realistic, distributed, simulated mission space that includes a synthetic environment, force representation, and behavioral representation. In other words NASM provides an air/space model to include the full spectrum of air warfare and those operations other than war in which uniformed personnel could be tasked.

Both NASM and its parent program JSIMS are intended to be fully distributed application/simulations permitting exercise participants to receive, process, and transmit commands and information across geographically dispersed locations. By implementing a distributed architecture, NASM (JSIMS) does not require the physical collocation of participants in wargaming exercises and the architecture accommodates the movement of command posts and units during exercises. The goal of a distributed simulation is for the audience to have a consistent perception of the entire battle space and this includes the graceful entry and exit of other actors/simulations at any point during the exercise. NASM will be capable of operating as a standalone model or integrated with JSIMS. Figure 1 depicts the top-level data and event flows into and out of NASM.

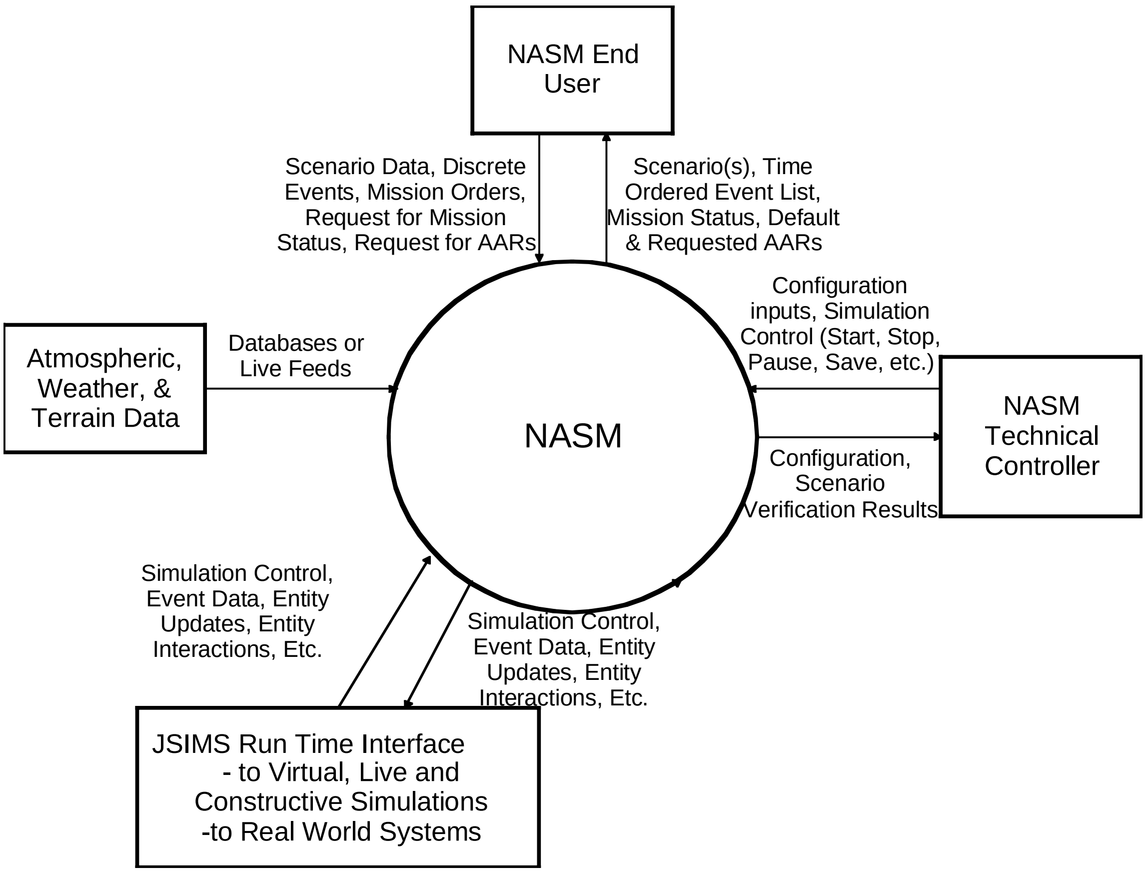


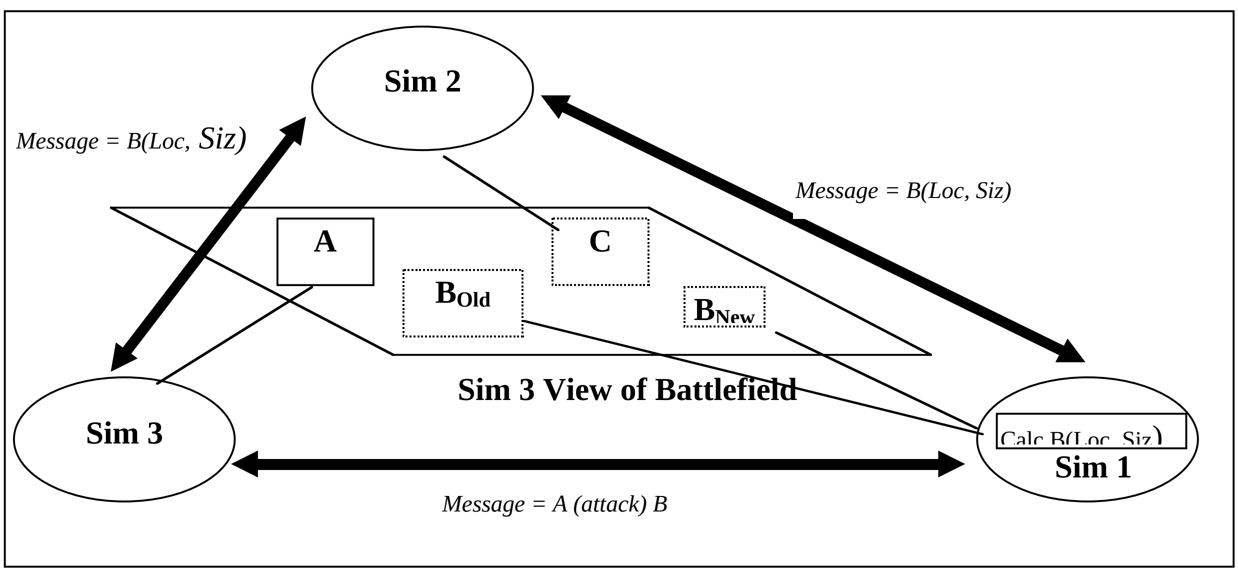
Figure 1. NASM to JSIMS Data and Event Flows

B.3 Distributed Architecture

The distributed architecture of NASM/JSIMS accommodates a multi-user, multi-platform environment where users are simultaneously executing applications while cognizant of other scenario attributes. The computing environment of JSIMS and NASM is a network with a core simulation engine (executable code) loaded on all platforms. In the case of NASM this would be a Local Area Network (LAN) and in the case of JSIMS this would be a Wide Area Network (WAN). The data/models are also distributed, but there is no core database, rather the data is dispersed among all the machines and each simulation engine is responsible for finding the necessary information to remain a valid participant/actor in the confederation. If data can not be located during a scenario then default data may be substituted to keep the exercise current.

JSIMS development relies heavily on object oriented software concepts, and entities such as airbases, aircraft, spacecraft, missiles, weapons, sensors and communication nodes have specific characteristics that determine their behaviors against which a status can be reported over the network throughout the exercise. JSIMS uses data classes to differentiate subclasses, which have significantly differing behavior. For example, missiles are split into surface-to-air, tactical air-to-surface, air launched cruise missiles, ground launched cruise missiles, theater ballistic missiles, and intercontinental ballistic missiles. Characteristics within each sub- class include sufficient detail to represent their actual performance during a scenario. For example aircraft attributes include the parameters of speeds (max, min, optimal, preferred) weapons, fuel capacity, fuel consumption as a function of speed, load, altitude, maintenance and support requirements. JSIMS also represents higher level objects, such as an airbase, with characteristics and capabilities inherited from subordinate objects, such as runways, shelters, communication facilities, aircraft, and personnel. These inventories of objects and their attributes are distributed amongst the many players in an exercise. It should be noted here that the Defense Modeling and Simulation Office (DMSO) High Level Architecture (HLA) is, among other things, an object oriented approach to M&S from which JSIMS incorporates models of weapon systems and uses part of the HLA common simulation engine. Stated simply, JSIMS is DMSO HLA compliant.

In JSIMS a distributed database of simulated object attributes is maintained where attributes are either public or private. A private attribute of an object is kept internally within a single simulation and is not the concern of the JSIMS interface protocol. Public attributes on the other hand are shared among all simulations. An object's public attributes are those properties of that object required by other simulations to interact with it. In Figure 2 the hypothetical battlefield contains combat units that have only two public attributes: location and size. The set of



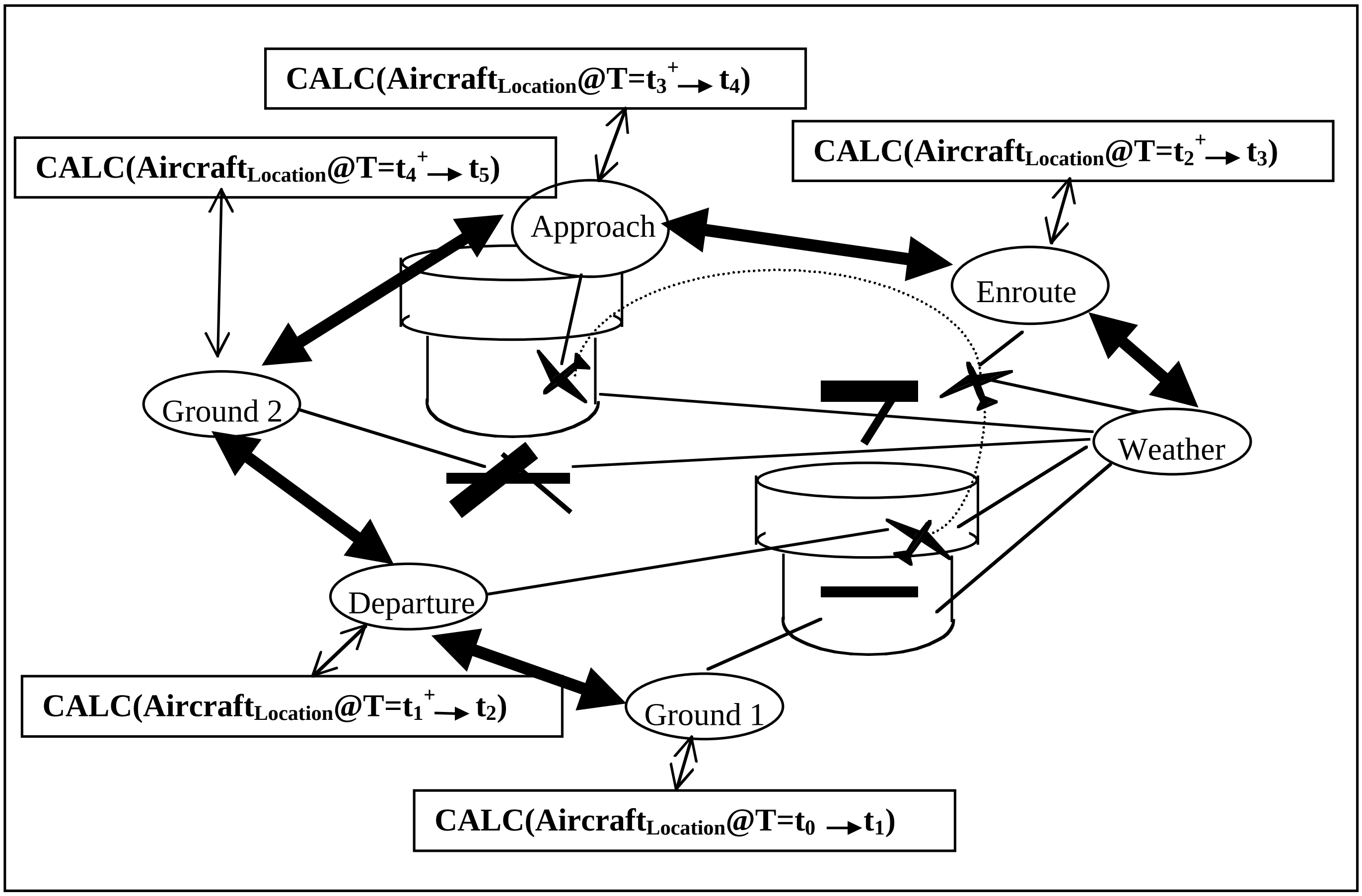
Note: Bold Arrows represent simulation connection interfaces

Figure 2. Example of Simulated Object Interaction and Attributes

objects are divided among three simulations. Each simulation is responsible for computing new values of the public attributes of its battlefield units and communicating these changes to other simulation. The other simulations use these values to project the existence of foreign objects into their simulation. These projections called “ghosts” allow a simulation to account for units in its calculations that are really controlled elsewhere. Assume a combat unit, A, owned by Simulation 3 encountered the ghost of an enemy unit, B, controlled by Simulation 1. Simulation 3 would send a message to Simulation 1 informing it that unit A is attacking unit B and send messages giving all other simulations new values for unit B’s public attributes. If B sustained sufficient damage it would change in size and possibly move to another location. This example demonstrates the two classes of exchange between simulation: public attribute maintenance and simulated object interaction.

The attribute ownership scheme of Figure 2 demonstrates a relatively simple, although constricting, way to manage public and private attributes. In the Figure 2 example, all attributes (Location and Size) of a single object (Unit B) were controlled by a single simulation (Sim 1). In general it is more desirable to establish object-attribute pairs and dynamically allocate ownership of attributes throughout run time. Again, referencing Figure 2, it would be possible to have Unit A’s location computed in Simulation 3 while its size determined by Simulation 1.

The usefulness of such a flexible approach is demonstrated using Figure 3. Here six independent simulations are joined to form a larger, seamless simulation that covers a larger scope than was intended by the original designers. Simulated aircraft move through the various phases of a flight while the individual simulations coordinate their control using a standard interface protocol. There are two kinds of simulated objects, airports and aircraft, each with a location and weather attribute. The Weather Simulation updates the value of the weather attribute for all objects. Airports and aircraft will have weather attribute update based on changes in the location and the advance of time. A Ground Simulation computes each airport’s operation, including taxiing of aircraft. The Ground Simulation updates the location attribute of aircraft as it simulates their motion on the airport surface. Once airborne, the aircraft’s location is computed by the Departure Simulation. To do this, the Departure Simulation observes changes in the location attribute of the taxiing aircraft. When an aircraft clears the runway, the Departure Simulation requests a change in attribute ownership from the Ground Simulation. The Ground Simulation passes the ownership of the particular aircraft’s location attribute to the Departure Simulation and stops computing new location values for it. Although the Ground Simulation is no longer involved in the simulation of the departing aircraft’s location, it could continue to display the aircraft’s ghost on its local area radar by listening to location updates generated by the Departure Simulation.



Note: Bold Arrows represent simulation connection interfaces

Figure 3. Example of Dynamic Distribution of Attribute Ownership

This section has offered some insight into how the DoD is addressing a distributed computing environment. The two examples were chosen to demonstrate the benefits and challenges of operating in a distributed architecture, be it in a stand-alone (multi-users, one simulation) or a federation (multi-users, multi-simulations). The added complexities of joining pre-existing simulations although difficult are not insurmountable because of object-oriented techniques and participants utilizing DMSO HLA standards, libraries and reusability features.

B.4 Time Management

JSIMS simulation times must be coordinated so that the times for all simulations in the confederation appear the same to users and so that event causality is maintained. The goal is to have all events occur in the same sequence in all simulations. All simulations under JSIMS must respect event causality, which is defined as processing messages in logical time order. By assigning event time stamps based on logical precedence, event causality is achieved by executing events in increasing time stamp order. This condition is frequently referred to as being temporal consistent or event consistent. There are two types of event consistency: internal or local consistency and global consistency. JSIMS assumes local consistency for each simulation participating in a confederation is correct. Global consistency is the property that participating simulations continue to exhibit local consistency in the presence of events introduced by interactive simulations. JSIMS supports a distributed time management strategy to guarantee global event consistency.

The goal of JSIMS time management is to reduce the effects of anomalies, which cause the simulated world to deviate from the real world in undesirable ways. For instance, cause-and-effect relationships in the real world may be distorted resulting in an execution where the "effect" appears to happen prior to the "cause." In most implementations, this added functionality will usually come at a cost, e.g. higher latency, higher network bandwidth consumption, and additional computation resources.

There are essentially two related problems that need to be addressed by JSIMS:

1. Events don't happen during the simulation execution when they are supposed to occur.
2. Events don't happen in the order that they are supposed to occur.

The first problem may be due to computers and networking infrastructure not being fast enough. The second problem can be solved, providing each participating simulation can precisely define the correct order of events. The approach of JSIMS time management is to enable complete solution of the second problem and at the same time enable the solution to the first to the extent that it is possible.

Central to the JSIMS time management services are mechanisms to order messages that are passed to all participating simulations. A variety of services are provided to support interoperability among simulations with diverse requirements. Five ordering mechanisms be currently specified in DMSO HLA: receive, priority, causal, causal and totally ordered, and time stamp order. These provide, in turn, increased functionality but at increased cost. JSIMS interactions with NASM use the time stamp message order (TSO) service. Messages utilizing this service adhere to either a conservative or optimistic protocol. JSIMS supports both protocols.

Conservative TSO messages will be delivered to simulations in time stamp order. JSIMS also ensures that no message is delivered to a federate "in its past", i.e. no message is delivered that contains a time stamp less than the simulation's current logical time. This is accomplished by forcing a simulation to explicitly request advances in logical time using JSIMS interface time advance services. The JSIMS interface will not grant the time advance request until it can guarantee that no messages containing a time stamp smaller than the time of the grant will later be received. The conservative TSO protocol insists that no simulation may schedule an event with time stamp less than the actor's current time plus a value L, then the JSIMS interface can allow concurrent delivery and processing of messages in a time window L time units wide beginning at the minimum logical time of any simulation. This value L is referred to as the lookahead for the simulation because it must be able to "look ahead" L time units into the future, or in other words, predict attribute updates and interactions at least L time units "ahead of time". Lookahead may, in general, be difficult to incorporate into certain classes of simulations, but nevertheless is very important for actors requiring guaranteed message ordering services to achieve acceptable performance.

Lookahead is clearly very intimately related to details of the simulation model, and thus cannot be determined automatically by the JSIMS interface. Some examples of where lookahead may be derived are described below.

- *Physical limitations concerning how quickly one simulation can react to an external event.* Suppose the minimum amount of time for a tank to respond to an operator's command (e.g., to fire ordnance) is 500 milliseconds. This means the federate can guarantee that it will not schedule the results of any new operator actions until at least 500 milliseconds into the future, providing a lookahead of this amount.
- *Physical limitations concerning how quickly one simulation can affect a second.* Suppose two tanks are ten miles apart, and there is also a maximum speed of a projectile fired from one tank to another. These constraints place a lower bound on how much time must elapse for the first tank to affect the second. Thus, events such as a projectile exploding at the second tank can be scheduled into the future, providing some degree of lookahead.
- *Tolerance to temporal inaccuracies.* Suppose a simulation produces an event at time T, but the receiver of the message for that event cannot distinguish between the event occurring at time T and T+100 milliseconds (e.g., in a training simulation, it might be the case that a muzzle flash occurring at time T is indistinguishable from one at time T+100 milliseconds). Then, the federate may schedule events 100 milliseconds into the future, providing a lookahead of this amount.
- *Time stepped simulations:* In a time stepped simulation, the lookahead is normally the size of the time step. This is because a federate can only schedule events into the next time step (or later), but not into the current time step.

Lookahead can change dynamically during the execution. However, lookahead cannot instantaneously be reduced. At any instant, a lookahead of L indicates to the JSIMS Runtime Interface (RTI) that the federate will not generate any event (using time stamp ordering) with time stamp less than C+L, where C is the federate's current time. If the lookahead is reduced by K units of time, the federate must advance K units before this changed lookahead can take effect, so no events with time stamp less than C+L are produced. The JSIMS interface requires each federate to specify lookahead information if any events utilizing the time stamp ordered service are generated. Care must be taken in developing the federate to maximize lookahead, as this can significantly affect performance. A single lookahead value is designated by each federate. This value may change at runtime, but reductions in lookahead do not take effect immediately, as noted above. In general, lookahead may simply not be present in some simulations, due to the nature of the activities being modeled. This may require the simulation to resort to optimistic processing techniques to achieve acceptable performance.

In Figure 4, a timeline is used to demonstrate the TSO for a conservative protocol with two simulations in the federation using the JSIMS RTI. In this example Simulation 1 wishes to execute an internal event at time t=14. It first requests a time advance to t=14, which immediately sets the RTI simulation output time to t+Lookahead=15. Before the time advance can be safely granted, all events arriving externally from other simulations at time<=14 must be executed in order to preserve temporal consistency. The JSIMS RTI is responsible for delivering those events up to the simulation in time-stamp order for execution. As the events (updates) from Simulation 2 are passed up, Simulation 1 local time is advanced. The time advance request from Simulation 1 is not granted until the output time of every other simulation in the federation is >t. At $t=14^{+}$, the request is granted and Simulation 1 is permitted to advance to t =14.

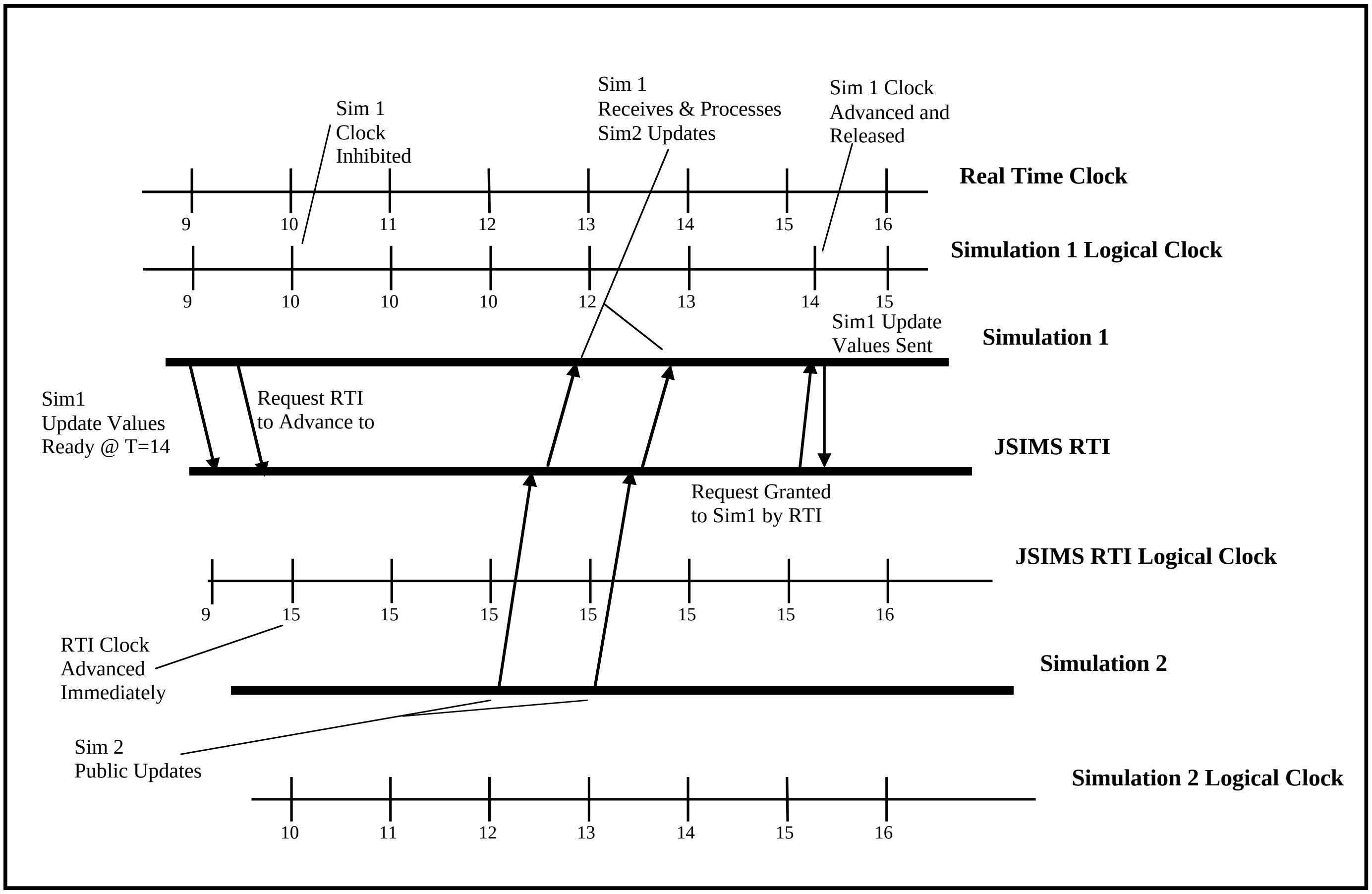


Figure 4. Example of a Conservative TSO Execution Scenario

Figure 4 attempts to demonstrate TSO delivery of messages using a conservative protocol to invoke the RTI synchronization mechanism, i.e. lookahead. This method guarantees messages will be delivered in logical-time order. The method may be equated to a worst case situation, because it anticipates any advance in simulation must first process events from other simulations. If there are no events to process, then the lookahead time was unnecessary. Clearly the disadvantages of this approach are institutionalized delays, and difficulty in handling concurrency that may result in large increases in execution speed. The alternative is the optimistic approach, which allows for messages to be processed out of order, but invokes a mechanism to correct the order. As with the conservative approach the optimistic protocol has each participating simulation maintaining its own logical clock and processes events from its local event list in strict simulation time order. Yet unlike the conservative approach the local clock will not be inhibited if some relative clock drift occurs. If another simulation posts and event earlier than the local clock a “time fault” is said to occur. The mechanisms invoked by the JSIMS RTI to correct the fault are “rollback” and “message cancellation”.

Rollback is the resetting of the simulation clock back so that it can correctly process the “out of order” message. Message cancellation is the technique used to nullify any events that the simulation posted before the arrival of the out of order message. With the optimistic time scheme the larger the clock drift occurring before a time fault, the more intense the processing power will be needed to maintain synchronization, and in a distributed environment enough bandwidth will be necessary avoid message saturation. For these reasons the selection of the optimistic TSO as the protocol for time management usually requires parallel microprocessor platforms.

The NASM Operational Requirements Document (ORD) states that playback and repeatability of simulations must be supported. The NASM ORD expands upon the requirement by declaring re-execution is repeatable if all conditions, e.g. timing and links, are the same and the results are identical to the previous simulation execution. Also the NASM ORD reads that if the simulation runs twice with the same inputs supplied at exactly the same times, including seeds from random elements, the simulation results should be identical. Given these conditions the only time management scheme to support these particular requirements is TSO, any other would permit events to occur in a non-deterministic manner, which would be insufficient to produce repeatable results.

B.5 Data Management

The selection of a TSO time management scheme can solve the problem where events are not occurring in a logical order. To get events to happen when they are supposed to occur, the optimal TSO can make some contributions, but there are other techniques available through data management that may not necessarily solve the problem, but can certainly make significant improvements to performance. The common denominator to all of these techniques is that each, in some way, attempts to minimize the data that needs to be transferred. Whether the data are objects, updates, events or interactions the goal is to prevent data processing or communication links from introducing penalizing delays. The selection of reliable communication protocols offering various choices of bandwidth is one approach to avoiding message saturation. For example NASM has selected TCP/IP running over dedicated T-1 lines offering 1.5 Mb/sec as their choice for WAN links. Locally, NASM simulations will be run with Asynchronous Transfer Mode offering 45 Mb/s of bandwidth. When NASM is to join other simulation in a JSIMS federation it is highly unlikely these other simulation packages will have such large amounts of bandwidth available, thus another technique available to each simulation, including NASM, is to “break up” their data into distributed relational or object-oriented databases. This would involve an efficient and highly effective way of partitioning data onto an array of computer servers. In this way, when data is transmitted (or received) much smaller amounts of data will need to be sifted through by the network, which therefore cuts down the amount of time to locate and return a result.

Simulations in a confederation intuitively broadcast global attributes and values to all participants. In a small confederation this may be acceptable, but as the number of simulated objects increases so do their attributes, resulting in an exponential increase in data flow. Inevitably the execution performance becomes unacceptable. The JSIMS RTI supports a more efficient data distribution to facilitate explicitly sending messages to only interested participants, similar to the computer network approach to solving the problem of message saturation by a technique called multi-casting (sending messages to only a defined subset of the network). Essentially it is a complex filtering technique utilizing the creation of areas of interest or routing spaces.

A routing space is a multi-dimensional coordinate system in which all participating simulations express an interest in either receiving or sending data to particular regions of the JSIMS participants (federation). Regions in a multi-dimensional routing space do not necessarily map to geographic region. For example while a location in space has 3 dimensions:

$$L(x, y, z),$$

a radio channel "space" could be composed of 6 dimension variables:

$$R(x, y, z, W, f, t)$$

where x, y, z are signal range; W is signal strength (watts); f is frequency; and t is time of broadcast. To use routing spaces, each federation defines the allowable routing spaces for the federation execution, including the dimensions (variables) of the routing space. Routing spaces are then initialized in the federation with a name and the number of dimensions. Routing spaces are then used by the federates to specify the distribution conditions for the specific data they are sending or expect to receive. Each federate decides which portions of those routing spaces are useful to it and specifies the logical areas of interest by bounding the dimensions of the selected routing space. The federate then uses these regions to specify conditions under which it…

- expects to reflect attribute values and receive interactions,
- will provide updates of attribute values and send interactions.

For example a radio channel space could contain hundreds of signals posted (published) from enemy and friendly aircraft, ships and units, but a distant federate might only have interest (subscribe to) in a unique set of signals using the radio channel coordinates: R(±10mi, ±10mi, ±10mi, >1000watts, >10KHz, <0500 hrs).

In summary the creation of routing spaces and assigning filtering characteristics to those spaces facilitates limiting information transmitted to only those messages of interest in order to reduce (1) the data set required to be processed by the receiving federate and (2) the message traffic over the network (LAN or WAN).

Appendix C. Modeling And Simulation Development Using High-Level Architecture (HLA)

This section provides a discussion of the HLA (High Level Architecture) developed by the DoD to support distributed simulations and to connect simulation entities. CHATS is similar in many aspects to DoD simulation efforts which HLA was targeted for.

The paper introduces HLA, discusses distributed simulation issues and how HLA would fit into CHATS, presents examples of FAA and other non DoD applications of HLA, discusses benefits and costs, provides a technical description, and a recommendation.

C.1 HLA Description

What is HLA?

The High-Level Architecture (HLA) establishes a common high-level simulation architecture to facilitate the interoperability of all types of models and simulations among themselves and with live systems, or live players. HLA is designed toward standardization in the M&S community and to facilitate the reuse of M&S components.

The HLA provides an infrastructure to couple components into a larger simulation. These components can be simulated or live. This supports the following:

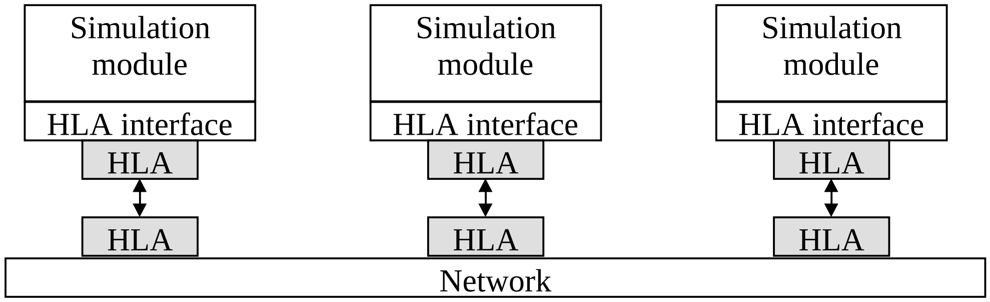
- coupling simulations into a larger simulation
- creating modular simulations
- creating distributed simulations
- use of data feeds from live systems in a simulation
- coupling simulations with live systems
- providing simulated and live data feeds into a system under test
- management over the simulation as a whole
- instrumenting the simulation

HLA provides a ready solution to implement a simulation where players are geographically dispersed and coupled with a network.

Where does HLA fit in to a simulation design?

HLA provides a medium to couple together elements of a distributed simulation. It provides a library which contains a rich set of services which allow a simulation to interoperate with other simulations. It provides a layer which lies over the network protocol (IP or UDP) which provides distributed simulation services. HLA improves interoperability and allows reuse of legacy systems.

HLA is not a simulation methodology. It is a simulation interface technology. The figure below depicts how HLA would fit into a simulation implementation.



Distributed simulation issues

The desire to implement a distributed simulation connected with a wide area network or the Internet brings about a host of issues. Some of potential issues are listed below. HLA provides services to handle these sorts of issues.

- How do I initiate the simulation?
- How do I coordinate startup of all the parts of a distributed simulation?
- How do I shut down the simulation in an orderly manner?
- What if a player is not present during startup?
- What if a player wants to join later?
- What if a player wants to quit?
- What if a player gets disconnected?
- How is unreliable message delivery handled?
- How is the flow time managed and kept synchronized among distributed players? How is causality handled (i.e. A causes B; A is reported to C before B is)
- How do you trade guarantee of message delivery against network bandwidth consumption?
- How do I deal with entities, which do the same thing in different region? From what point is a distributed simulation managed and how?
- How do I handle who owns what entity?
- How do I handle who can make changes to an entity? How do I handle reporting of an event?
- How do I handle transfer of ownership of an entity, both pull and push?
    - Do you own X? Can I have X? etc.
- How do I add or delete entities? How do I notify relevant players of this action?
- How do I handle the flow of time and events among a set of different players?
- How do I handle the flow of time if different players handle time differently?
- How do I handle the flow of time between real and simulated components?

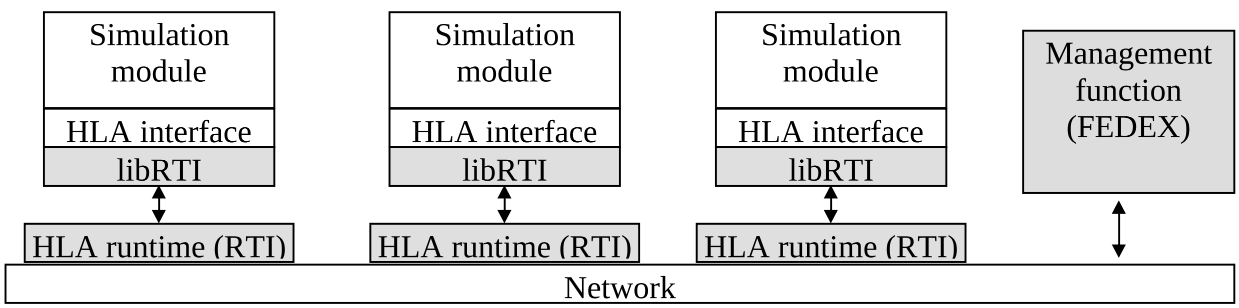
HLA vs. internal simulation workings

HLA does not drive the design of the inner workings of a simulation. The figure above depicts a HLA interface added to each simulation module. HLA does drive the design of the interface to a simulation, not the design of the simulation itself. In practice, the simulation modules often are legacy simulations. In these cases, an HLA interface is added to the existing simulations.

The interface may be implemented in a number of ways. It can be added to the simulation code, implemented as middleware, or implemented by the addition of a gateway machine. A gateway machine is employed in situations where simulations are hosted on peculiar hardware, like some flight simulators. This approach was used in the JFAN project described later.

The HLA runtime environment provides an interface called RTI Ambassador to each component of the simulation. A library, LibRTI provides the application programmers interface to couple to the RTI Ambassador. As an analogy, X-Windows, or the MS-Windows GDI provides an application programmers interface which lets a program draw graphics in windows on a screen shared with other program's windows. LibRTI provides an application programmers interface to implement an interface between elements of a distributed simulation. In all these cases, pre-existing run time code handles the implementation details.

The figure below expands on the previous figure by naming the HLA components and depicting the HLA management function which will be described later. The shaded portions are HLA provided components.



HLA is object oriented. It uses object oriented interface design, and its libraries are object oriented (C++, JAVA or Ada). This does not demand that the simulation implementation be object oriented. As an example HLA is being incorporated into existing Joint STARS simulation modules which are primarily FORTRAN.

HLA permits coupling between systems which employ different hardware including the PC and workstations from Digital (Alpha), Sun, IBM, Silicon Graphics (SGI) and Hewlett-Packard (HP). Coupling across different operating systems such as HP-UX, AIX, Solaris, NT, Win95, Linux and IRIX is supported.

Where would HLA fit into CHATS?

HLA would provide the coupling between CHATS components. It supports the environments of its principal components, NARIM which uses HP-UX and ETMS which has been recoded in C to run on standard Unix.

The HLA Interface definition tools provide a common language to define interfaces, and provides a methodology to change them. In a HLA based design, HLA interface definition tools would be used in the design process to define interfaces. The same tools support making changes to interfaces as required. These tools provide a vehicle to define interfaces for the designers, and provide data files which describe the components of the simulation for the HLA environment.

The Federation Executive (FEDEX) is an HLA process which provides distributed simulation management functions. FEDEX provides a console interface for manual operations. The FEDEX would provide some of the functionality required of the operations center for CHATS.

A HLA implementation will allow the simulation to run either in a distributed mode of operation, using the Internet or a wide-area network, or in a colocated mode using an Intranet or a local-area network.

HLA would provide a modular CHATS implementation. This implementation would have the flexibility to change and grow as needs arise. This flexibility would reduce the difficulty of adding other entities to the simulation, and would allow incorporating parts or all of CHATS in other simulation activities.

C.2 Civilian Uses of HLA

JFAN

JFAN (Joint FAA/Army/NASA Federation) involved the development of a HLA based simulation. This effort involved the Army Advanced Modelling and Simulation Facility, the FAA National Simulation Capability (NSC), and NASA Ames Research Center. Motivations to use HLA differed between these organizations. The DoD mandate provided the Army's initial reason to use HLA. The FAA NSC intended to review viability of distributed simulation technologies and methodologies. With the advent of HLA the FAA/NSC determined that distributed modeling technology is reaching a level of maturity which could be adopted for the NSC program. The NASA Ames goal was to evaluate the costs and capabilities associated with an HLA implementation.

The JFAN experience with HLA was generally positive. Some of the problems encountered were a consequence of using an early and incomplete release of HLA (version 1.0.3; 1.3 is current). The parties involved would in retrospect stay with the choice of HLA. Positive observations include: the ability to integrate diverse systems ("good for bridging organizations"), ease of making design and architectural changes late in the project ("...adding/deleting information is surprising painless"), and the positive impact of HLA's interface definition methodology ("...helps identify interface requirements of each participant early in the project.").

Issues include: HLA's steep learning curve, performance over marginal network links, and the need for some software and/or hardware upgrades. HLA's interface definitions do not cover low level data formatting.

The current version address most of the problems encountered in the JFAN implementation. The current version is the first to implement all of the services in the interface specification.

Other examples

A MITRE funded project is tasked to integrate the HLA and a commercial gaming product developed by Digital Image Design (DID). DID is a UK based company and is known for its quality flight simulations (TFX, EF2000 v2.0, & F22). MITRE is currently working with them to provide HLA compatibility with their new gaming engine. Refer to the site: http://www.did.com

The Training Department, Gagarin Cosmonaut Training Centre 141100, Star City, Moscow Region, Russia is using HLA to interconnect Space simulators and simulations in Russia and in Western Europe in the framework of the International Space Station development program.

Web21 Inc. is using the HLA RTI now for interoperability of simulations of both physical and logical domains at various levels of definition. This is a business-to-business training application where the operator reacts to fairly abstract alerts. Web21 plans to continue to comply with the HLA as the implementation evolves. By next year, human face animation and body animation behaviors will be added to reach the entertainment market. Web21 plans to ask set-top box manufacturers to support HLA.

C.3 Benefits and Costs

Risk mitigation

HLA provides ready facilities to develop modular simulations. This mitigates risk in implementing a distributed simulation in several ways:

Use of a technology intended to address the problem at hand. HLA is intended to implement simulations like what is envisioned for CHATS. Using a tool intended for the job is less risky than designing your own.

Divide and conquer. HLA is intended to allow simulations to be implemented in pieces. If problems are encountered in the implementation of a part of the simulation, impacts are confined to those parts. The mechanism which integrates the pieces, is off-the-shelf, and certainly more mature than a home-grown implementation.

Mitigates integration risks. HLA in intended to integrate simulation components which are designed by different organizations. It forces early agreement of data exchange among participants. It also makes later changes to the data exchange model simple to implement. For most DoD applications, HLA simulation components are developed by different companies in geographically disparate locations.

A need to combine existing simulations with different implementation approaches commonly arises. As an example, different simulations may handle the flow of time differently. In many cases legacy simulations are involved. HLA is intended to couple simulations for these sorts of cases.

As mentioned before, use of HLA minimizes the chance of failing to properly implement the integration infrastructure, which ties the pieces of the project together. If the integrating infrastructure doesn't work, the whole project doesn't work.

Eases changes in design. HLA also provides a structured process and tools to define the interface between simulation components. The tool is the Object Model Development Tool (ODMT). This promotes early interface definition (mitigating late changes). This does not imply that making late changes is difficult. Quite the opposite is true; making later changes to the data exchange model is simple. This ability to make changes also simplifies adding new components or changing the scope of the simulation.

Learning curve issues

HLA does present a substantial learning curve. This creates a scheduling issue with getting the requisite training in place in time. The impact of this learning curve is probably less than the impact of development of an in-house implementation. Providing early training of an in-house expert is recommended. The DoD Defense Modeling and Simulation Office (DMSO) provides free HLA training in the Washington DC area. DMSO also provides a help desk.

This learning curve appears similar to learning enough Macintosh Toolbox, or Microsoft Windows Application Programming Interface (API) to write a simple program. HLA and the previous two examples present rich APIs and require learning a significant number of API calls to get started. Programmers comfortable with working with APIs of similar scope to graphical user interface (GUI) APIs (and associated event handling, callbacks etc.) encounter less of a challenge.

One example is a programmer used to a command line environment (no GUI or event driven experience) who cites a 6 month learning curve with no help from individuals with HLA experience. This programmer was able bring other programmers up to speed in less time. Another individual remarked that a thorough understanding of how HLA works was a worthwhile investment, which simplified implementation.

Obtaining HLA

HLA documentation, software, libraries and tools materials can be obtained from the Web site http://www.dmso.mil/hla. Software can be downloaded after registering. There is no cost.

C.4 HLA Technical Description

HLA definitions

HLA has a jargon associated with it. It is necessary to define several terms associated with HLA before describing it.

- Federation - a simulation built up of modular components or federates
- Federate - a simulation, live system interface, or support utility
- Run Time Interface (RTI) - Supporting infrastructure which couples federates in a federation

HLA elements

The HLA is defined by three elements. These components are listed below. A common language is specified to describe the attributes and actions associated with an object. An interface specification specifies services which support interactions between components. A set of rules ensures proper interaction between components. These elements are described below:

Object Model Template (OMT)	The OMT provides a common language for documenting and recording information describing an HLA component. OMT data is entered and manipulated with software tools which present the data in both human readable and machine readable forms.
Interface Specification	The interface specification identifies how federates will interact with the federation and, ultimately, with one another. The specification is divided into six management areas, which will be explored in more detail in this document. These six areas are - Federation Management — Controls an exercise - Declaration Management — Negotiates data exchange - Object Management — Communicate entity existence and characteristics - Ownership Management — Controls permissions to change object attributes - Time Management — Coordinates flow of time - Data Distribution Mgt — Routes information
HLA Rules	The Federation Rules describe the responsibilities of federates and their relationships with the RTI. There are ten rules. Five relate to the federation and five to the federate.

Each of these HLA elements will be treated in greater detail.

Object Model Template

Object models provide an identification of the set of objects, which represent the real world for a specific application. This identification includes: object characteristics, attributes, relationships, and behaviors.

There are three significant types of object models:
- Federation Object model - Contains all shared information used in a federation of simulations.
- Simulation Object Model - Describes salient characteristics of a simulation, and presents objects and interactions, which can be accessed externally
- Management Object Model - Identifies objects and interactions used to manage a federation.

The methodology for building each of these object model types, and the formats, are the same.

These object models are built with a software utility used to gather the necessary information. This object model information is accessible in human readable and machine-readable form. The machine-readable output is used to tell the HLA runtime infrastructure about the simulation and its components. This utility is the Object Model Development Tool (ODMT). ODMT also provides consistency checking and validation functions.

Building these object models is a prerequisite to running a simulation in the HLA run time infrastructure. The process of developing these object models requires that the entire system be considered to determine such things such as the description of the data communicated between simulations, conditions of data update and various other information that is pertinent to the specification of a simulation system for interoperability purposes.

Interface Specification

The HLA Interface Specification partitions the exchanges that take place between federates and Federation into six management areas. A library of routine calls is associated with each of these areas. A HLA interface is built into a simulation using routine calls which fall into these categories. A short description of each of the six areas follows.

<u>Federation Management</u>

Federation management includes such tasks as creating federations, joining federates to federations, observing federation-wide synchronization points, effecting federation-wide saves and restores, resigning federates from federations, and destroying federations.

<u>Declaration Management</u>

Declaration management includes publication, subscription and supporting control functions. Federates that produce objects (or object parts) or that produce interactions must declare exactly what they are able to publish (i.e., generate). Federates that consume objects (or object parts) or that consume interactions must declare their subscription interests.

The Declaration Management Services of the RTI uses publication and subscription information (declared by federates participating in a federation) to throttle the data placed on the network. Control signals issued by the RTI can be used to constrain type registration and instance updates. The RTI effectively serves as an intelligent switch - matching up producers and consumers of data, based on declared interests and without knowing details about the data format or content being transported.

Object Management

Object management includes instance registration and instance updates on the object production side and instance discovery and reflection on the object consumer side. Object management also includes methods associated with sending and receiving interactions, controlling instance updates based on consumer demand, and other miscellaneous support functions.

Ownership Management

The RTI allows federates to share the responsibility for updating and deleting object instances with a few restrictions. It is possible for an object instance to be wholly owned by a single federate. In such cases, the owning federate has responsibility for updating all attributes associated with the object and for deleting the object instance. It is possible for two or more federates to share update responsibility for a single object instance. When update responsibility for an object is shared, each of the participating federates has responsibility for a mutually exclusive set of object attributes. Only one federate can have update responsibility for an individual attribute of an individual object at any given time. In addition, only one federate has the privilege to delete an object instance at any given time.

Time Management

The focus of time management is on the mechanics required to implement time management policies and negotiate time advances.

The HLA accommodates a variety of time management policies. The RTI provides optional time management services to coordinate the exchange of events between federates. Events can be associated with a point in time and the RTI can assure causal behavior. It is also possible for one or more federates in a federation to ignore time all together. By default, the RTI does not attempt to coordinate time between federates. One strength of the HLA is that it not only supports a variety of time management policies, but also anticipates interoperability between federates with different policies. Even if the optional time management services are ignored, it pays to understand available time management schemes.

In a federation, time always moves forward. However, the perception of the current time may differ among participating federates. Time management is concerned with the mechanisms for controlling the advancement of each federate along the federation time axis. In general, time advances must be coordinated with object management services so that information is delivered to federates in a causally correct and ordered fashion.

In some situations, it is appropriate to constrain the progress of one federate based on the progress of another. In fact, any federate may be designated a regulating federate. Regulating federates regulate the progress in time of federates that are designated as constrained. In general, a federate may be "regulating," "constrained," "regulating and constrained," or "neither regulating nor constrained." By default, federates are neither regulating nor constrained. The RTI recognizes every federate as adapting one of these four approaches to time management. A federation may be comprised of federates with any combination of time management models. That is, a federation may have several federates that are regulating, several federates that are constrained, or several federates that are regulating and constrained.

A federate that becomes "time regulating" may associate some of its activities (e.g., updating instance attribute values and sending interactions) with points on the federation time axis. Such events are said to have a "time-stamp." A federate that is interested in discovering events in a federation-wide, time-stamp order is said to be "time constrained." The time management services coordinate event exchange among time-regulating and time-constrained federates. Such coordination levies certain rules on participants.

Once again, federates are neither time regulating nor time constrained by default. The activities of these federates are not coordinated (in time) with other federates by the RTI. Such federates need not make use of any of the time management services. However, these federates may participate in a federation where time-stamped events are exchanged. It is important to understand how time-stamped events are perceived by federates that are not constrained. Conversely, it is important to understand how events generated by a non-regulating federate are perceived by a constrained federate.

Data Distribution Management (DDM)

DDM provides a flexible and extensive mechanism for further isolating publication and subscription interests - effectively extending the sophistication of the RTI's switching capabilities. In DDM, a federation "routing space" is defined. The *routing space* is a collection of "dimensions." The *dimensions* are used to define "regions." Each *region* is defined in terms of a set of "extents." An *extent* is a bounded range defined across the dimensions of a routing space (it represents a volume in the multi-dimensional routing space).

HLA Rules

The HLA rules are presented below. They present significant information on the nature of HLA.

Federation Rules:	Federate Rules:
Federations shall have an HLA Federation Object Model (FOM), documented in accordance with the HLA Object Model Template (OMT).	Federates shall have an HLA Simulation Object Model (SOM), documented in accordance with the HLA Object Model Template (OMT).

In a federation, all representation of objects in the FOM shall be in the federates, not in the run-time infrastructure (RTI).	Federates shall be able to update and/or reflect any attributes of objects in their SOM and send and/or receive SOM object interactions externally, as specified in their SOM
During a federation execution, all exchange of FOM data among federates shall occur via the RTI.	Federates shall be able to transfer and/or accept ownership of attribute dynamically during a federation execution, as specified in their SOM.
During a federation execution, federates shall interact with the run-time infrastructure (RTI) in accordance with the HLA interface specification.	Federates shall be able to vary the conditions (e.g., thresholds) under which they provide updates of attributes of objects, as specified in their SOM.
During a federation execution, an attribute of an instance of an object shall be owned by only one federate at any given time.	Federates shall be able to manage local time in a way which will allow them to coordinate data exchange with other members of a federation.

HLA architecture

The HLA rules hint at the underlying architecture. Each federate (simulation component) contains an interface referred to as a RTI Ambassador. A federate communicates to another federate only through its RTI Ambassador. Any RTI Ambassador couples the federate with the RTI software, which runs in each machine. The RTI handles communications over the network, and represents a layer, which lies on top of the IP or UDP network protocol. HLA also requires that one instance of the FEDEX is running. FEDEX provides a central management function for the RTI.

Any network can be used, ranging from the internet to internal networks or dedicated links. A network is not necessary to couple federates; several federates or all federates can run in one machine. Changes in distribution of federates among machines or network configurations has a negligible impact.

The RTI Ambassador interface is unique to each simulation. It is built with the application programmers interface contained in the HLA interface specification. A legacy simulation would require the addition of an interface to the RTI Ambassador. The RTI and the FEDEX are off-the-shelf.

Network Implementation

Running distributed simulations over the Internet is often mentioned in discussing HLA. The term ‘Internet’ does not necessarily mean the public Internet. Military applications often involve classified information; this information can’t be sent over public networks. The military implements its own closed Internets. An example is SIPRNET (Secret Internet Protocol Routing NETwork) which provides secure connectivity.

Several examples of simulation specific Internets are the Defense Simulation Internet (DSI) and the Technical Interoperability Network (TIN). The TIN couples the Modelling and Simulation Center (MASC) with the Command and Control Unified Battlespace Environment (CUBE). The MASC provides simulation, and the CUBE incorporates live systems or systems under test. MASC/CUBE activities combine simulated forces, test range activity, live activity, and simulated systems. The MASC and the CUBE are not colocated.

Military distributed simulations are often coupled with wide area networks which support the internet protocol. These networks are implemented as an infrastructure to provide connectivity between the sites involved. Security and robustness are drivers for this choice.

Is HLA another ADA?

Use of a product with government or military origins is sometimes questioned. The Ada programming language experience is an example which is often brought up. The approach associated with introduction and development of HLA differs from some of the previous history. Some observations on this follow.

Ada was mandated by the DoD, but was not supported. At the time of the mandate, Ada development tools were immature, expensive, and worked poorly. The complexity of Ada put an excessive burden of the computer resources available at the time.

The developers of HLA are providing the support necessary to propagate this standard. Instead of mandating a standard with little more than theoretical basis, the DoD contracted out for the development of HLA tools and is passing them out for free. Free training and support is made available. HLA does not have viable competitors in the commercial marketplace. The current DoD emphasis on employment of commercial-off-the-shelf solutions would have resulted in the selection of a product if such a product were available. Despite the newness of HLA, usage outside the DoD sphere exists. This usage includes the FAA, Germany and Britain. HLA is submitted as an IEEE standard.

HLA was developed to address a pervasive compatibility problem present in DoD simulations. DoD wants and needs to do simulation in the large, and the existing infrastructure did not deliver. HLA was developed to address that problem.

C.5 Recommendation

Use of HLA deserves strong consideration when considering pursuit of simulation development.

The use of HLA's structured process for building the object models is strongly encouraged. Piecing together interface definitions in a non-structured manner invites omission of critical data. Having a clear process to define interfaces allows identification of data exchange requirements for each participant early in the design cycle.

HLA provides modularity and scalability. The impact of design changes is reduced, and allows an application of a whole implementation or reuse of pieces of an implementation for other purposes.

HLA's design embodies a lot of experience and thought about building distributed simulations. Several individuals remarked that deviating from the recommended HLA way of doing business was a mistake. This was not a consequence of a HLA problem, but a result of missing design issues which the HLA methodology would have surfaced. HLA forces the developer to think through issues early in the design process which would later turn into problems.

The use of FEDEP, HLA's structured process for building the object models is strongly encouraged. Piecing together interface definitions in a non-structured manner invites omission of critical data. Having a clear process to define interfaces allows identification of data exchange requirements for each participant early in the design cycle. Individuals working the FAA JFAN project initially skipped FEDEP, and later wished they hadn't.

Before embarking on development of an architecture of a simulation, forming an understanding of the services performed by the HLA API is strongly recommended. The HLA RTI Programmers Guide provides a easily understood overview. The problems which various parts of the HLA API are intended to address should be understood before development of a distributed simulation design.

Latency and capacity are two significant performance issues. JFAN involved coupling real time in-cockpit flight simulation. Interest exists in the entertainment industry for using HLA for multi-player video gaming. These sorts of applications demand significantly better latency performance than CHATS will. However, the large number of entities which CHATS will involve may drive capacity. For this reason, investigating capacity is advised.

Even if HLA is not used as a part of the CHATS implementation, following the process defined by HLA, and understanding the programmers guide to the HLA API will contribute to a better simulation design.

Appendix D. CHATS Cost Estimate

Table D-1 presents a detailed cost estimate for CHATS. Note that over 90% of the cost is the expert labor required to develop and run the system.

General Assumptions

- The Simulation Policy Team and the Airspace and Rules Team are Federal employees who will be paid for by their agencies.
- There is a cost escalation of 5% per year for salaries and services.

Specific Assumptions

The following assumptions are discussed by step and item number from Table D-1.

Item B3. The prototype will have 3 workstations co-located at the central complex, one for the simulation operator, one for the traffic management team and one for an airline team. These are assumed to cost $5K each loaded with appropriate operating systems and software. In addition, the LAN will have a server whose cost is assumed at $10K. Other costs of network services, storage devices and miscellaneous brings the estimate up to $40K.

Item B4. It is assumed that NASA will host the physical space of the central operations complex at no cost to the project.

Item B5. The software development needed is to modify and adapt existing software systems, namely NARIM and ETMS, to the needs of the simulation.

Item B7. It is assumed that two individuals, one representing traffic management and one representing an airline, are needed half-time each. If these individuals are currently employed by the stakeholders, their salaries for this job may be picked up by their employers under other accounts.

Item C3. It is assumed that two individuals representing traffic management and an airline are needed half-time each during the prototype simulations.

Item D3. New workstations are needed to be located at the FAA's System Command Center and at an assumed 3 AOCs, making 4 in all. At $5K each this comes to $20K, and miscellaneous costs of $10K at the sites brings the figure to $30K.

Item D4. Based on a recent SRC investigation, a 256K connection to the Internet with a leased server and managed firewall from an ISP, utilizing existing T1 communication service (which is probably available at NASA), will cost $60K a year plus $5K installation cost.

Item D5. It is assumed that the new required space at the System Command Center and at the AOCs will be made available at no cost to the project.

Item D6. This is an estimate of the software development needed to expand from the localized to the distributed simulation.

Item E4. It is assumed that during the 1 year conduct of simulations, a half-time labor commitment is required from each of the four player sites, for a total of 2 labor-years.

Table D-1. CHATS Detailed Cost Estimate

Development Step	Duration (Months)	Item #	Resources	Labor Level	Amount[1]	Cost Per Unit ($K)	Cost ($K)
A. Prototype Scope, Concept, and Plan	2	1	Planning and system engineering personnel	2	0.33	$160	$53
B. Develop Using Rapid Prototype Process	10	1	Planning and directing personnel	1	0.83	$160	$133
		2	System engineering personnel	0.5	0.42	$160	$67
		3	Hardware, platforms, networks, displays		3 platforms, LAN+server		$40
		4	Physical space				free
		5	Software developers	0.5	0.42	$150	$63
		6	Test and operations personnel	0.5	0.42	$130	$54
		7	Stakeholder evaluation personnel	1	0.83	$160	$133
		8	Documentation personnel	0.1	0.08	$120	$10
			Subtotal		3.00		$500
C. Conduct Simulations Using Prototype	12	1	Planning and directing personnel	0.5	0.50	$168	$84
		2	Test and operations personnel	0.5	0.50	$137	$68
		3	Stakeholder personnel participating as players	1	1.00	$168	$168
		4	Documentation personnel	0.15	0.15	$126	$19
			Subtotal		2.15		$339

Development Step	Duration (Months)	Item #	Resources	Labor Level	Amount[1]	Cost Per Unit ($K)	Cost ($K)
D. Complete CHATS Development	12	1	Planning and directing personnel	0.5	0.50	$168	$84
		2	System engineering personnel	0.5	0.50	$168	$84
		3	Hardware, platforms, networks, displays		4 platforms		$30
		4	Internet server & service		1 year	$60 + $5 install	$65
		5	Physical space				free
		6	Software developers	0.5	0.50	$158	$79
		7	Test and operations personnel	0.5	0.50	$137	$68
		8	Stakeholder evaluation personnel	1	1.00	$168	$168
		9	Documentation personnel	0.1	0.10	$126	$13
			Subtotal		3.10		$591
E. Conduct Full-Scale Simulations	12	1	Planning and directing personnel	1	1.00	$176	$176
		2	Internet server & service		1 year	$63	$63
		3	Test and operations personnel	0.5	0.50	$143	$72
		4	Stakeholder personnel participating as players	2	2.00	$176	$353
		5	Documentation personnel	0.2	0.20	$132	$26
			Subtotal		3.70		$690
Total					12.28		$2,173
[1] For labor entries, "Amount" is measured in labor-years.							